



**Universidade de Brasília**

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

# Otimização de desempenho computacional do sistema de análise de imagens faciais PhotoQuality

Jônathas Alves Oliveira

Monografia apresentada como requisito parcial  
para conclusão do Curso de Computação — Licenciatura

Orientador  
Prof. Dr. Flávio Vidal

Brasília  
2015

Universidade de Brasília — UnB  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Curso de Computação — Licenciatura

Coordenador: Prof. Dr. Wilson Veneziano

Banca examinadora composta por:

Prof. Dr. Flávio Vidal (Orientador) — CIC/UnB  
Prof. Dr. Alexandre Zaghetto — CIC/UnB  
Prof. Dr. André Costa Drummond — CIC/UnB  
Prof. Dr. Bruno Luigi Macchiavello — CIC/UnB

### **CIP — Catalogação Internacional na Publicação**

Oliveira, Jônathas Alves.

Otimização de desempenho computacional do sistema de análise de imagens faciais PhotoQuality / Jônathas Alves Oliveira. Brasília : UnB, 2015.

81 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. análise de desempenho, 2. desempenho, 3. ajuste fino de *software*, 4. validação de imagens faciais digitais, 5. otimização, 6. teste de *software*, 7. computação distribuída

CDU 004

Endereço: Universidade de Brasília  
Campus Universitário Darcy Ribeiro — Asa Norte  
CEP 70910-900  
Brasília-DF — Brasil



# Otimização de desempenho computacional do sistema de análise de imagens faciais PhotoQuality

Monografia apresentada como requisito parcial  
para conclusão do Curso de Computação — Licenciatura

Prof. Dr. Alexandre Zaghetto    Prof. Dr. André Costa Drummond  
CIC/UnB                                  CIC/UnB

Prof. Dr. Bruno Luigi Macchiavello  
CIC/UnB

Prof. Dr. Wilson Veneziano  
Coordenador do Curso de Computação — Licenciatura

Brasília, 16 de julho de 2015

# Dedicatória

Dedico este trabalho a minha família e amigos que sempre me ofereceram apoio para realizar este projeto e me motivaram desde o início para prosseguir uma disciplina após a outra, até chegar a este Trabalho de Conclusão de Curso. Dedico também aos meus professores que me forneceram o embasamento necessário, ao longo de todos esses anos, para que eu fosse capaz executar um projeto com este da melhor maneira possível.

# Agradecimentos

Agradeço primeiramente a minha família pelo incentivo que deram para que eu fosse capaz de realizar todos os meus sonhos até aqui, incluindo concluir o curso de Licenciatura em Computação da Universidade de Brasília. Agradeço também ao professor Flávio Vidal ter estado presente nos momentos que precisei do seu auxílio sempre disposto á oferecer as orientações necessárias para a realização deste trabalho de maneira adequada. Agradeço também ao **Cespe-Cebraspe** por ter me oferecido a oportunidade de trabalhar neste projeto e ter me concedido os meios para que fosse possível a realização deste trabalho. Por fim, agradeço a própria Universidade de Brasília por ter me proporcionado um curso do qual me orgulho bastante e que me dá a segurança de que possuo as competências e habilidades necessárias para ser um profissional bem visto no mercado.

# Resumo

Este trabalho se propõe a analisar o problema de desempenho em um sistema recém-desenvolvido e denominado **PhotoQuality** voltado para a análise de imagens faciais digitais. A solução das questões relacionadas ao desempenho insuficiente é feita através de ajustes finos e alteração no modelo de execução do sistema. No decorrer do projeto, são abordados conceitos de análise de desempenho e teste de *software*, computação concorrente, acesso á dispositivos de entrada e saída, além de ligações estáticas e dinâmicas. As melhorias são apresentadas e discutidas a partir dos valores de vazão medidos e qualificados. O sistema de análise de imagens será utilizado como mecanismo de verificação da identidade de pessoas, com vistas a evitar fraude em processos de seleção em concursos públicos, vestibulares e demais avaliações.

**Palavras-chave:** análise de desempenho, desempenho, ajuste fino de *software*, validação de imagens faciais digitais, otimização, teste de *software*, computação distribuída

# Abstract

This study aims to analyze the performance problem in a newly developed system called **PhotoQuality** focused on analysis of digital facial images. The solution of issues, related to the underperformance, is done through fine adjustments and changes in the system execution model. During the project will be covered performance analysis concepts and software testing, concurrent computing, access to input and output devices, as well as static and dynamic linkages. Improvements are presented and discussed from the measured and qualified throughput values. The image analysis system will be used as a verification mechanism of the identity of persons, in order to prevent fraud in the selection process in public procurement, exams and assessments.

**Keywords:** tuning, throughput, response time, distributed computing, *software* testing, software performance analysis

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa . . . . .	2
1.2	Definição do problema . . . . .	2
1.3	Objetivos . . . . .	3
1.3.1	Objetivo geral . . . . .	3
1.3.2	Objetivo específico . . . . .	3
1.3.3	Observações importantes . . . . .	3
1.4	Estrutura do trabalho . . . . .	4
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>5</b>
2.1	Introdução . . . . .	5
2.2	Avaliação do desempenho . . . . .	5
2.2.1	Técnicas de avaliação . . . . .	6
2.2.2	Abordagem sistemática . . . . .	7
2.2.3	Definição dos requisitos de avaliação do desempenho . . . . .	7
2.3	A técnica da mensuração e suas ferramentas . . . . .	8
2.3.1	Carga de trabalho . . . . .	8
2.3.2	Monitores . . . . .	10
2.3.3	Métricas . . . . .	11
2.4	<i>Tuning</i> de desempenho . . . . .	12
2.5	Execução de múltiplos processos . . . . .	12
2.6	O processo de <i>swap</i> . . . . .	14
2.7	Bibliotecas estáticas e dinâmicas . . . . .	15
<b>3</b>	<b>Padrões de qualidade para imagens faciais</b>	<b>16</b>
3.1	O Documento nº 9.303 <i>ICAO</i> de 2006 . . . . .	16
3.1.1	Relevância da imagem facial para o e-passaporte . . . . .	18
3.2	Passaporte biométrico (e-passaporte) . . . . .	19
3.2.1	Funcionamento do passaporte biométrico . . . . .	20



3.2.2	Propriedades particulares do e-passaporte . . . . .	20
3.3	Padrões <i>ICAO</i> para fotos do e-passaporte . . . . .	20
3.3.1	Posicionamento/enquadramento do sujeito . . . . .	21
3.3.2	Pose . . . . .	22
3.3.3	Qualidade da imagem facial . . . . .	23
3.3.4	Esquema de cores e iluminação . . . . .	23
3.3.5	Penteado e expressão facial . . . . .	23
3.3.6	Luminosidade e fundo . . . . .	24
3.3.7	Óculos . . . . .	24
3.3.8	Chapelaria . . . . .	24
3.3.9	Especificação para crianças . . . . .	26
<b>4</b>	<b>Sistema de análise de imagens faciais</b>	<b>28</b>
4.1	O subsistema <i>PhotoQuality</i> . . . . .	28
4.1.1	Finalidade do projeto . . . . .	29
4.1.2	Processo de análise automática de imagens . . . . .	29
4.1.3	Análise da qualidade de imagens digitais . . . . .	29
4.1.4	Deteção de Indivíduos por imagens . . . . .	30
4.1.5	Desenvolvimento . . . . .	30
4.1.6	Responsáveis pelo projeto . . . . .	30
4.1.7	Características e Funcionalidades . . . . .	32
4.2	O subsistema <i>PSI</i> . . . . .	34
4.2.1	Módulos . . . . .	35
4.2.2	Características . . . . .	35
4.3	Infraestrutura . . . . .	36
<b>5</b>	<b>Avaliação e estratégias para a otimização do desempenho</b>	<b>39</b>
5.1	Avaliação da utilização dos componentes físicos . . . . .	40
5.1.1	Utilização dos dispositivos de E/S . . . . .	42
5.1.2	Utilização da memória RAM . . . . .	44
5.1.3	Utilização do processador . . . . .	45
5.2	Hipóteses para o baixo desempenho . . . . .	46
5.2.1	Ausência de mecanismos de concorrência . . . . .	46
5.2.2	Leitura em disco . . . . .	47
5.2.3	Execução de um processo a cada nova requisição . . . . .	47
5.3	Soluções para o problema de desempenho . . . . .	48
5.3.1	Estratégias de implementação dos ajustes . . . . .	48

<b>6</b>	<b>Apresentação e avaliação dos resultados</b>	<b>53</b>
6.1	Cenário I – Medições de desempenho iniciais . . . . .	54
6.2	Cenário II – Ajuste dos parâmetros de instanciamento e concorrência . . .	56
6.2.1	Conclusão ajustes . . . . .	57
6.3	Cenário III – Execução do <b>PhotoQuality</b> como biblioteca dinâmica . . . .	58
6.3.1	Conclusão dos ajustes . . . . .	59
6.4	Cenário IV — Gravação e leitura das imagens em <i>RAM disk</i> . . . . .	60
6.4.1	Conclusão dos ajustes . . . . .	62
<b>7</b>	<b>Conclusões</b>	<b>64</b>
7.0.2	Trabalhos futuros . . . . .	65
	<b>Referências</b>	<b>67</b>

# Lista de Figuras

2.1	O processo de cíclico de <i>tuning</i> de sistemas [1]	13
3.1	Especificação para enquadramento e posicionamento de indivíduos em imagens contidas em documentos de viagens [2]	22
3.2	Especificação da pose do indivíduo em imagens contidas nos documentos de viagem [2]	22
3.3	Especificação de qualidade de imagens contidas nos documentos de viagem [2]	23
3.4	Especificação de penteado e expressão facial do indivíduo em imagens contidas nos documentos de viagem [2]	24
3.5	Especificação de luminosidade e plano de fundo das imagens contidas nos documentos de viagem [2]	25
3.6	Especificação para a utilização de óculos por indivíduos em imagens contidas nos documentos de viagem [2]	25
3.7	Especificação para a utilização de chapelaria por indivíduos em imagens contidas nos documentos de viagem [2]	26
3.8	Especificação captura de imagens faciais de crianças [2]	26
4.1	O processo de chamada do subsistema <b>PhotoQuality</b>	32
4.2	Fluxograma de funcionamento do subsistema <b>PSI</b>	36
4.3	Processo de análise de uma imagem	37
4.4	Diagrama da infraestrutura de execução da plataforma	38
5.1	Volume de transferência de dados dos dispositivos de entrada e saída em kilobytes.	42
5.2	Quantidade de operações de E/S por segundo.	43
5.3	Utilização de memória RAM.	44
5.4	Utilização do processador.	45
5.5	Esquema de acesso ao hardware pelo <b>c++</b> nativo e pelo gerenciado [3].	51
6.1	Médias de vazão por amostra e seus erros estimados, dentro do Cenário I.	55
6.2	Médias de vazão por amostra e seus erros estimados, no Cenário I e II.	57

6.3	Médias de vazão por amostra e seus erros estimados, no cenário II e III. . .	59
6.4	Médias de vazão por amostra e seus erros estimados no cenário III e IV. . .	61
7.1	Médias de vazão por Cenário, utilizando-se as observações com 1.600 imagens.	65

# Lista de Tabelas

2.1	Cr�terios para Sele��o das T�cnicas de Avalia��o [4]	7
4.1	As atividades envolvidas em cada etapa do projeto	31
5.1	Rela��o entre dispositivos avaliados e os contatadores utilizados com suas defini��es.	41
6.1	Resultados dos testes desempenho no Cen�rio I	55
6.2	Resultados das vaz�es obtidas no Cen�rio I e no Cen�rio II	56
6.3	Resultado do teste T com duas amostras e varian�as presumidamente diferentes, utilizando as observa��es com 12.800 imagens.	58
6.4	Resultados das vaz�es obtidas no cen�rio III e no cen�rio II	60
6.5	Resultado do teste T com duas amostras e varian�as iguais, utilizando as observa��es com 12.800 imagens.	60
6.6	Resultados das vaz�es obtidas no cen�rio IV e no cen�rio III	62
6.7	Resultado do teste T com duas amostras e varian�as presumidamente diferentes, utilizando as observa��es com 12.800 imagens.	62

# Capítulo 1

## Introdução

As possibilidades de desenvolvimento de aplicações têm passado por constantes evoluções no decorrer das últimas décadas. Isso proporcionou o surgimento de um grande número de alternativas de ferramentas e métodos para o desenvolvimento de sistemas computacionais para atender aos requisitos de que a indústria necessita [5].

No entanto, esse fato não significa que as tecnologias que surgiram inicialmente estejam sendo completamente suplantadas pelas novas. Isso tem gerado uma necessidade crescente de desenvolvimento de metodologias de integração de aplicações completamente distintas de uma forma que trabalhem em conjunto de maneira performática.

Este trabalho se propõe a estudar e otimizar o desempenho de um sistema de análise de imagens, cujas plataformas caracterizam-se por possuir distinções quanto às plataformas de desenvolvimento utilizadas. Além de terem sido desenvolvidos em plataformas diferentes, estão distribuídos em diferentes servidores. O ponto central é a definição e implementação de estratégias para a otimização do sistema de análise de imagens. Além disso, deverão ser efetuadas avaliações de desempenho que deem suporte para a definição algumas hipóteses que expliquem o desempenho insuficiente observado e, dessa forma, poder efetuar os ajustes de maneira eficaz.

O sistema a ser avaliado e otimizado consiste em uma plataforma de análise e validação de imagens faciais. Esse sistema é composto por um servidor Web e outro encarregado de analisar as imagens que são enviadas pela *Internet*. A camada de *software* responsável pela análise das imagens propriamente dita consiste em um sistema denominado **PhotoQuality**. Essa camada é desenvolvida na plataforma **C++**, utilizando a biblioteca *OpenCV*. Este sistema não contém interfaces que possibilitem ser acessado pelo servidor *Web*, por meio da rede local. Devido a isso, foi desenvolvida mais uma camada de *software* responsável por prover uma interface por meio da rede que possibilite a qualquer outro dispositivo da rede consumir os serviços de análise de imagens. Essa camada adicional foi denominada **PSI**.

O PSI, além de prover a interface de acesso pela rede local, é um serviço que implementa uma linguagem de comunicação simplificada, entre aplicações distribuídas, que possibilita flexibilidade e interoperabilidade. Essa linguagem é denominada *Web Services Description Language (WSDL)*. O *WSDL* é um formato *XML* para descrever serviços de rede como um conjunto de nós de extremidade que operam em mensagens que contenham qualquer informação orientado a um documento ou processo. Essa linguagem é extensível e permite a descrição dos nós de extremidade e suas mensagens, independentemente do formato ou protocolos de rede utilizados na comunicação [6].

Diferentemente do *PhotoQuality*, o serviço PSI foi desenvolvido na plataforma *.NET*, utilizando-se a linguagem *c#*, de modo que a complexidade reside justamente na integração destas duas tecnologias. Integração essa que, feita de maneira inadequada, poderá impactar o desempenho geral do sistema.

## 1.1 Justificativa

Segundo Molyneaux [7], antes de ser disponibilizado em produção, um *software* deve ser submetido a testes de desempenho e escalabilidade sob situações de utilização adversas. Caso o sistema entre em produção sem os devidos testes que avaliem seu desempenho, esse sistema poderá apresentar problemas de lentidão em situações em que há uma elevada taxa de utilização. Um sistema executado de maneira lenta poderá impactar a usabilidade do próprio sistema, o que poderá trazer prejuízos para a reputação desse sistema e até mesmo trazer consequências para a imagem da empresa proprietária do sistema, podendo resultar numa redução da utilização, pelos clientes, dos serviços prestados pela empresa.

As consequências, caso esse sistema continue a funcionar com o desempenho atual, podem ser desastrosas para a instituição proprietária do sistema, já que o processo de cadastro e gerenciamento das contas dos usuários poderá acarretar em aumento na quantidade de chamados à central de atendimento, elevando os custos operacionais. Além disso, podem fazer com que os usuários também desistam de utilizar os serviços providos pela instituição, o que pode causar a perda de novos interessados em contratar os serviços futuramente, devido a uma experiência desagradável com um dos sistemas.

## 1.2 Definição do problema

O problema a ser abordado consiste nas médias de vazão, que foram consideradas insuficientes pelos proprietários do sistema. Essa avaliação leva em consideração a vazão observada preliminarmente (cerca de duas imagens por segundo), em detrimento da ca-

pacidade computacional disponibilizada exclusivamente para a realização das atividades de análise de imagens faciais.

Um conjunto de hipóteses poderá ser levantado para explicar a causa do problema. As hipóteses mais prováveis (as quais este trabalho se dedicará a investigar) são: sobrecarga de utilização de um ou mais dispositivos, que poderá afetar também o desempenho dos demais; ausência de um ajuste fino voltado para o aumento da vazão; interferência dos demais processos que estão em execução na máquina.

## 1.3 Objetivos

### 1.3.1 Objetivo geral

Estudar um sistema de análise de imagens preexistente e estabelecer estratégias para o aumento de seu desempenho.

### 1.3.2 Objetivo específico

Os objetivos do trabalho consistem em:

1. Estudar o modelo de funcionamento do sistema;
2. Realizar medições de desempenho;
3. Encontrar hipóteses que expliquem seu baixo desempenho observado;
4. Aplicar possíveis soluções por meio de cenários de ajuste;
5. Obter uma vazão média de (no mínimo) 5,4 imagens por segundo, devendo chegar até 14 imagens por segundo, em situações de alta demanda;
6. Comparar resultados em cada um dos cenários. (*benchmarking*).

### 1.3.3 Observações importantes

Vários autores, em especial Patterson et. al., definem que o desempenho pode ser atribuído a duas medidas distintas, *throughput* (ou vazão) e tempo de resposta. O tempo de resposta, também chamado de tempo de execução, é o tempo total necessário para o computador realizar uma determinada tarefa. Essa tarefa poderá incluir acesso a disco, acesso à memória principal, tarefas que demandam operações de entrada e saída (E/S), sobrecarga do sistema operacional, tempo de execução da *CPU* e assim por diante. Já a *vazão* é uma medida de desempenho que permite obter o número de tarefas completadas por uma determinada unidade de tempo [8].



No contexto deste trabalho, o termo “desempenho” será referente exclusivamente aos índices de vazão. Sendo assim, o tempo de resposta é uma métrica que não será levada em consideração dentro do escopo do projeto de otimização do desempenho do sistema de análise de imagens. Essa opção foi feita apenas objetivando a simplificação das atividades a serem realizadas no decorrer do projeto. Além disso, o proprietário do sistema deseja apenas elevar a quantidade de imagens que podem ser analisadas em um segundo.

Apesar dos objetivos do projeto serem o teste e a otimização do desempenho dos sistema de análise de imagens **PhotoQuality**, os ajustes a serem efetuados ocorrerão, em sua maioria, no subsistema **PSI**, de modo que o sistema de análise de imagens seja tratado como uma caixa preta. Dessa maneira, objetiva-se que os testes e ajustes realizados no decorrer do trabalho possam ser aplicados a demais sistemas que efetuam outras tarefas, que podem não ter relação com processamento de imagens digitais.

## **1.4 Estrutura do trabalho**

O trabalho está estruturado em oito capítulos. No capítulo 2, apresenta-se a revisão teórica que embasará este trabalho. No capítulo 3, introduz-se o padrão de documentos de passaportes legíveis por computador. No capítulo 4, os dois sistemas envolvidos são explicados: o de análise de fotos e o de disponibilização da interface de acesso. No capítulo 5, é feito o levantamento das hipóteses do que está causando o desempenho insuficiente do sistema, sendo realizado um levantamento das hipóteses e o estabelecimento de um conjunto de medidas para solucionar o problema. No capítulo 6, apresentam-se os resultados obtidos após a aplicação de cada uma das soluções encontradas, além da comparação com os resultados anteriores, quando a solução funcionava com desempenho desfavorável. No capítulo 7, apresentam-se as conclusões e os trabalhos futuros.

# Capítulo 2

## Revisão Bibliográfica

Este capítulo faz um apanhado dos conhecimentos fundamentais que tratam da avaliação e *tuning* de desempenho de um sistema. Para auxiliar na introdução do tema, o capítulo começa abordando sistematicamente a avaliação do desempenho. Em seguida, é feita uma seleção de técnicas de métricas voltadas para a medição de desempenho. Em seções posteriores, são abordadas as técnicas de mensuração do desempenho e as ferramentas necessárias para a realização da atividade.

### 2.1 Introdução

Segundo a definição atribuída pelo *IEEE Standard Glossary of software Engineering Terminology* [9], desempenho seria o grau com o qual um sistema ou componente realiza sua função designada dentro de certos parâmetros, tais como velocidade, precisão ou utilização de recursos.

A definição de Smith [10] estabelece que: Desempenho refere-se à capacidade de resposta, pode ser o tempo necessário para responder a eventos específicos, ou ainda o número de eventos processados em um determinado intervalo de tempo. Portanto, performance é definida como o atributo computacional que descreve a qualidade do serviço fornecido por um determinado sistema, em termos de requisitos não funcionais no âmbito de desempenho.

### 2.2 Avaliação do desempenho

A avaliação do desempenho objetiva prever o comportamento do sistema de uma maneira quantitativa. Quando um novo sistema de computador deve ser construído ou um sistema existente deve ser reconfigurado, adaptado ou ter seu desempenho aprimorado, a avaliação

de desempenho do sistema deve ser feita para se prever o impacto que as modificações arquiteturais, ou de implementação, poderão introduzir em todo o sistema [11].

A avaliação do desempenho diz respeito a qualificar o serviço entregue por um computador ou sistema de comunicação. É útil para determinar o tempo de reposta, comparar sistema que despenham serviços semelhantes, dentre outras aplicações [12].

É desejável avaliar o desempenho do sistema para assegurar que ele é adequado para as aplicações desejadas e que é rentável para ser desenvolvido, ou se existe fisicamente, ele pode ser ajustado para prover um desempenho desejável sobre dadas restrições de recursos e condições operacionais [11].

Na seção seguinte, são apresentadas as principais técnicas de avaliação de sistemas. Este trabalho utilizará apenas uma das técnicas. Porém, para dar um panorama geral do assunto, todas as técnicas serão abordadas.

### 2.2.1 Técnicas de avaliação

Existem três técnicas para avaliação de desempenho. São elas: mensuração, simulação e modelagem analítica [4].

- **Mensuração** – Técnica mais fundamental, é necessária até mesmo nas demais técnicas, para calibrar os modelos desenvolvidos. Algumas mensurações efetuadas podem atingir melhores resultados em *hardware* ou *software*. Uma vez que há diversos fatores incontroláveis (ou controláveis) no experimento de mensuração, os dados coletados devem ser analisados por técnicas de estatísticas para se chegar a conclusões significativas [13].
- **Simulação** – Envolve a construção de um modelo para o comportamento do sistema, conduzindo-o com uma apropriada abstração de carga de trabalho [13].
- **Modelagem Analítica** – Envolve a construção de um modelo matemático do comportamento do sistema (sob um nível desejável de detalhamento) e a resolução do mesmo [13].

### Critérios para seleção da técnica de avaliação

A consideração chave na decisão da técnica de avaliação está no estágio do ciclo de vida ao qual o sistema está. Mensurações são possíveis somente se alguma implementação similar ao sistema proposto já existe. O que ocorre com frequência, por exemplo, no caso da concepção de uma versão melhorada de um produto preexistente. Se é um novo conceito, a modelagem analítica é a técnica que resta para ser escolhida [4]. A tabela 2.1 fornece um resumo dos prós e contras de cada uma das técnicas de avaliação apresentadas.

Tabela 2.1: Critérios para Seleção das Técnicas de Avaliação [4]

Critério	Mensuração	Modelagem Analítica	Simulação
1. Estágio	Pós protótipo	Qualquer um	Qualquer um
2. Tempo necessário	Varia	Pequeno	Médio
3. Ferramentas	Instrumentação	Analistas	Linguagens de computadores
4. Precisão	Varia	Baixa	Moderada
5. Manipulação dos parâmetros	Difícil	Fácil	Moderado
6. Custo	Alto	Baixo	Médio
7. Poder de convencimento	Alto	Baixo	Médio

### 2.2.2 Abordagem sistemática

Jain [4], apesar de destacar que os problemas de performance são exclusivos de cada sistema, define (de modo geral) os passos em comum para se avaliar o desempenho de sistemas. Esses passos consistem em:

1. Fixação de metas e definição do sistema;
2. Listagem dos serviços oferecidos pelo sistema, bem como seus resultados;
3. Seleção das métricas;
4. Listagem dos parâmetros de análise, bem como os fatores (parâmetros que sofrerão variação no decorrer da análise);
5. Seleção da técnica de avaliação;
6. Seleção da carga de trabalho;
7. Confeção dos experimentos a serem realizados;
8. Análise e interpretação dos dados;
9. Apresentação de resultados.

### 2.2.3 Definição dos requisitos de avaliação do desempenho

É desejável avaliar o desempenho de um sistema de computador, devido ao interesse que se tem de assegurar que um dado sistema é adequado para ser utilizado nas aplicações pretendidas. Além de se assegurar que ele satisfará dados requisitos de eficiência e flexibilidade, por meio das medições de desempenho de interesse [13].

Todos os sistemas devem ser projetados e operados com requisitos específicos desempenho em mente. É essencial que todos os requisitos de desempenho, de qualquer sistema a ser projetado, sejam indicados no início e antes de investir tempo e recursos financeiros nas fases finais do projeto, que incluem testes e implementação. Mesmo que as opções de avaliação do desempenho sejam amplas, os fundamentos permanecem os mesmos [11].

Para os requisitos serem significativos, deve-se especificar limites, se houver, sobre as configurações, cargas de trabalho e ambientes. Esses requisitos são inadequados caso eles tenham um ou mais dos problemas listados [4]:

- *Inespecífico*: Números claros não são especificados. Aplica-se a utilização de palavras qualitativas, como baixo, alto, raro e extremamente.
- *Imensurável*: Não há forma de medir o sistema e verificar se ele atinge os requisitos.
- *Inaceitável*: Os valores numéricos de requisitos, se especificados, são definidos com base no que pode ser obtido ou no que parece ser adequado. Se for feita uma tentativa de definir os requisitos de forma realista, eles acabam por ser tão baixos, que se tornam inaceitáveis.
- *Inatingível*: Muitas vezes, para parecerem bons, os requisitos são definidos muito além do que é possível se realizar.
- *Incompleto*: Não são feitas tentativas para a especificação de possíveis resultados.

## 2.3 A técnica da mensuração e suas ferramentas

A técnica da mensuração, utilizada na avaliação de desempenho computacional, é considerada a técnica mais verossímil; no entanto, ela é a mais dispendiosa. A mensuração, ou técnica do teste, pode ser implementada em sistemas reais ou em versões protótipo de um sistema que se deseja desenvolver. A mensuração do desempenho de um sistema envolve o monitoramento do sistema em estudo, enquanto esse está sujeito a um particular conjunto de cargas de trabalho ou pela aplicação de programas de *benchmark* [11].

### 2.3.1 Carga de trabalho

Independentemente de qual das três técnicas de avaliação foi selecionada, é necessário fornecer certos insumos para o sistema real ou modelo. Há duas categorias de insumos relevantes. A primeira categoria inclui parâmetros que podem ser controlados pelo desenvolvedor ou administrador do sistema e abrange rotinas de agendamento, interconexões entre dispositivos e políticas de alocação de recursos. A segunda categoria abrange

insumos gerados pelo ambiente ao qual o sistema está implantado que não podem ser controlados pelos responsáveis pelo sistema. Esses insumos podem ser, por exemplo, o intervalo de chamados a um serviço. Os insumos podem ser usados para colocar carga em um sistema real (ou um modelo de simulação) e assim, posteriormente, determinar distribuições para modelagem analítica ou simulação. Insumos são denominados cargas de trabalho. A caracterização da carga de trabalho é uma das questões centrais na avaliação de desempenho, pois nem sempre fica claro quais aspectos da carga de trabalho são importantes, quais os níveis de detalhamento em que a carga deverá ser registrada e como essa carga deverá ser representada e utilizada [13].

O termo carga de trabalho denota qualquer carga utilizado em estudos de desempenho. Uma carga de testes poderá ser real ou sintética. Uma carga real é uma observada em um sistema sendo utilizados em operações normais. Ela não poderá ser repetida, e, portanto, geralmente não é adequada para ser utilizada como carga de trabalho em testes. Por outro lado, uma carga de trabalho sintética possuem características similares às cargas reais e poderá ser utilizada repetidamente de maneira controlada. É desenvolvida e utilizada para estudos. A principal razão para utilizar uma carga sintética é que ela é uma representação ou modelo de uma carga real. Outras razões para utilizar cargas sintéticas são que os dados não são de um ambiente real e, portanto não são extensos ou contém dados sensíveis; a carga pode ser facilmente modificada sem afetar as operações. ela pode ser facilmente transportada para diferentes sistemas devido ao seu tamanho reduzido; e ela poderá possuir facilidades para mensuração embutidas [4].

### **Selecionando a carga de trabalho**

Com o objetivo de desenvolver uma carga de trabalho que possa ser considerada como sintética e, portanto, selecionada para testes de desempenho, Jain [4] considera importante levar em consideração alguns aspectos importantes na fase de seleção da carga de trabalho. Essas considerações são:

- *Serviços desempenhados pelo sistema:* Cada sistema provê um determinado número de serviços. É necessária a identificação dos serviços providos pela sistema em estudo, bem como os componentes sob estudo internos ao sistema. Assim, é possível determinar a carga mais apropriada para avaliar o desempenho do sistema ou de um dos seus componentes. Jain também pontua que se deve dar prioridade primeiramente à carga que permita avaliar o desempenho do sistema por completo e não apenas a um dos seus componentes.

- *Nível de detalhamento*: Para minimizar o tamanho da carga de testes, é possível utilizar algumas das estratégias voltadas ou para a redução de cargas extensas ou para aumentar o nível de detalhamento.
  1. Requisição mais frequente – Significativa quando há requisições frequentes cujas características se diferenciam das demais;
  2. Frequência dos tipos de requisição – Significativa quando determinadas requisições impactam o desempenho em grau muito mais elevado que outras;
  3. Cronometragem das sequências de requisições – Gera cargas muito vastas, mas é necessária, caso esteja medindo-se um sistema que demanda elevado detalhamento;
  4. Amostragem de demandas por recursos – Utilização de recursos por requisição;
  5. Distribuição das demandas por recursos – Há elevada variância de demanda por recursos.
- *Representatividade*: A carga selecionada deverá representar proporcionalmente os aspectos de tempo de entrada; demanda por recursos; sequência e quantidade que os recursos são demandados.
- *Atualização*: A carga deve acompanhar os padrões de utilização dos serviços.

## **Técnicas de caracterização da carga de trabalho**

A caracterização da carga é considerada uma questão importante na avaliação de desempenho, pois nem sempre são claros o nível de detalhamento que a carga deverá possuir, os aspectos da carga que são significativos e o método a ser utilizado para representar a carga de trabalho. Na caracterização da carga de trabalho, o termo “usuário” poderá não ser um ser humano. Nas literaturas relacionadas, é utilizado o termo “componente de carga” ou “unidade de carga”. Isso significa que a caracterização da carga de trabalho busca caracterizar um componente típico. As quantidades mensuradas, requisições e demandas por recursos utilizados para caracterizar a carga são chamados parâmetros. Os parâmetros de impacto significativos estão inclusos enquanto aqueles de impacto menor são geralmente excluídos [14].

### **2.3.2 Monitores**

Monitores são definidos como ferramentas utilizadas para observar e registrar as atividades de sistema em teste. As principais funções do monitor utilizado na avaliação do

desempenho são: observar o desempenho do sistema; coletar estatísticas de desempenho; analisar os dados coletados; exibir resultados, se possível [11].

Dependendo do mecanismo de ativação do monitor ele poderá ser classificado como orientado a evento ou orientado por temporização (monitor de amostragem). Monitores orientados a evento são ativados somente na ocorrência de certos eventos. Monitores de amostragem são ativados em intervalos de tempo definidos [4].

### Terminologias relevantes

Alguns termos que serão utilizados com frequência ao se operar monitores [4]:

- **Overhead:** Consumo de recursos devido a execução do monitor;
- **Taxa de Entrada:** Frequência máxima de eventos que o monitor é capaz de capturar;
- **Resolução:** Nível de precisão da carga, métrica ou análise das informações.

### Monitoramento por *software*

O monitoramento por software é mais adequado em mensurações em nível de programação e requer algum suporte do sistema operacional e do hardware. A facilidade mais importante é a cronometragem programável, necessária para controlar a duração dos períodos. Ele pode ser carregado em um intervalo desejável. Ao iniciar, um monitor em software inicia a contabilização e gera uma interrupção quando o tempo restante chega a zero. A rotina de tratamento da interrupção permite tomar ações apropriadas, tais como leitura do estado das variáveis ou processamento dos dados coletados, bem como o encerramento do experimento [13].

### 2.3.3 Métricas

Em cada estudo de desempenho, um conjunto de critérios de desempenho ou métricas deve ser escolhido. Uma forma de preparar esse conjunto é listando os serviços oferecidos pelos sistemas. Para cada requisição feita ao sistema, há diversas possibilidades de resultado. Para muitas métricas, o valor da média é tudo o que importa. No entanto, não se deve ignorar o efeito da variabilidade. Se duas métricas consistem essencialmente na mesma informação, é menos confuso estudar apenas uma [11].

#### Métricas de interesse

- A **Vazão** (ou *throughput*) é definida como a taxa (requisições por unidade de tempo) na qual as requisições podem ser atendidas pelo sistema. Para sistemas de processa-



mento de transações, a vazão é medida em Transações Por Segundo (TPS). A vazão geralmente aumenta quando a carga inicial cresce. A partir de um certo volume de carga, a vazão para de aumentar, sendo que (em vários casos) pode até mesmo diminuir. Frequentemente, o tempo de resposta poderá ser muito alto sob um valor elevado de vazão. Nesses casos, é mais interessante conhecer o valor máximo da vazão que não excede o valor máximo para o tempo de resposta preestabelecido. Isso poderá ser denominado **capacidade utilizável** [4].

- O **Tempo de Resposta** possui duas possíveis definições. Pode ser definido como o intervalo entre o fim do envio da requisição (pelo cliente) e o começo da resposta correspondente pelo sistema. A segunda definição diz que o tempo de resposta é o tempo entre o pressionamento do botão de enviar (pelo cliente) e o fim do recebimento do último *byte* da resposta. A segunda definição é preferível caso o comprimento da resposta seja longo [4].

## 2.4 *Tuning* de desempenho

*Tuning* do sistema refere-se à alteração de alguns parâmetros operacionais, de modo que todos os usuários experimentem uma melhor resposta do sistema, considerando-se que o *tuning* refere-se ao ajuste do programa de interesse somente para usuários individuais [13].

Jain [4] mantém uma visão mais ampla do assunto, atribuindo uma definição que não depende da natureza do sistema. Define *tuning* como o processo de ajuste de parâmetros do sistema, objetivando a otimização do desempenho do mesmo, independente de quem seja ou de qual a sua finalidade.

O propósito do *tuning* é otimizar o desempenho por meio da alteração de políticas de gerenciamento dos recursos. É necessário decidir quais parâmetros devem ser alterados e como alterá-los para se obter o máximo benefício potencial. A experimentação direta é uma técnica simples a se utilizar, mas poderá não ser adequada em ambiente de produção [13].

## 2.5 Execução de múltiplos processos

Sistemas multiprogramados proveem de um ambiente no qual vários recursos de hardware são utilizados eficientemente, mas não possibilitam interação direta do usuário com o sistema de computador. O compartilhamento de tempo (ou *multitasking*) é uma extensão lógica da multiprogramação. Em sistemas de tempo compartilhado, a CPU executa múltiplas tarefas por meio da alternância entre elas, mas a alternância ocorre em uma



Figura 2.1: O processo de cíclico de *tuning* de sistemas [1]

frequência elevada, de modo que os usuários continuam sendo capazes de interagir com vários programas, sem perceber a atuação do mecanismo [15].

Em vez de deixar a *CPU* na ociosidade, o sistema operacional vai rapidamente colocar o processo de algum outro usuário para ser executado. Em sistemas operacionais de tempo compartilhado, deve-se assegurar um tempo de resposta razoável, que é possível mediante o mecanismo em que o processo é completamente ou parcialmente tirado da memória principal, sendo colocado no disco. Isso permite que haja a execução de programas que não estejam totalmente em memória [15].

Um processo é uma unidade de trabalho do sistema. Um sistema consiste em uma coleção de processos, alguns dos quais são processos do sistema operacional e os demais são processos dos usuários. Todos esses programas em execução podem potencialmente funcionar concorrentemente, compartilhando uma única *CPU*. Para aumentar a utilização do processador, computadores devem manter diversos programas em memória, surgindo a necessidade de gerenciamento de memória. Atividades que são feitas pelo sistema operacional por meio de mecanismos de gerenciamento de memória [15]:

- Manter o registro de quais partes da memória estão sendo utilizados e por quem;
- Decidir quais processos (ou partes dele) e dados devem ser removidos da memória;
- Alocar e desalocar espaço da memória quando necessário.

Um processo (ou tarefa) pode querer carregar e executar outro programa. Muitos processos são desenvolvidos para ficarem em estado de espera, aguardando um chamado para iniciar o processamento, sendo despertados quando uma conexão é feita. O processo fonte da comunicação é denominado “cliente” e o processo que recebe a chamada é denominado “servidor” [15].

O estado do processo é definido em parte pela atividade atual desse processo. Um processo pode estar em um dos seguintes estados [15]:

- Novo. O processo está sendo criado.
- Executando. Instruções estão sendo executadas.
- Aguardando. O processo está aguardando algum evento ocorrer (como uma conclusão de I/O).
- Pronto. O processo está aguardando para assumir o processador.
- Terminado. O processo encerrou a execução.

Cada processo é representado no sistema operacional por um controle de bloco de processo que contém informações do processo, como: estado atual; contador de programa; registradores de *CPU*; informações do agendados de *CPU*; informações de gerenciamento de memória; informações de utilização de hardware e demais informações; status de I/O [15].

O objetivo da multiprogramação é ter sempre algum processo em execução, para maximizar a utilização da *CPU*. O objetivo do compartilhamento de tempo é a alternância de utilização do processador entre processos. Assim, os usuários podem interagir com diferentes programas. Para atingir esse objetivo, o agendador de processo seleciona um processo (normalmente entre um grande conjunto de processos disponíveis) para a execução na *CPU*. Para um sistema de processador único, não haverá mais que um processo em execução no processador. Se há mais processos, o resto estará aguardando a *CPU* ser liberada e reagendada [15].

Alternar a *CPU* para outro processos requer o salvamento do estado do processo atual e a restauração do estado do próximo processo. Essa tarefa é conhecida como troca de contexto. O sistema não realiza trabalho útil durante a troca de contexto, portanto o processador fica ocioso neste período. A velocidade desse procedimento varia de máquina para outra, dependendo de fatores físicos ou lógicos [15].

## 2.6 O processo de *swap*

Para ser executado, um processo deve estar em memória. Esse processo pode ser transferido temporariamente da memória para um repositório auxiliar e trazido de volta para

prosseguir com sua execução. O repositório auxiliar deve ser grande o suficiente para acomodar cópias de todas as imagens de memória. O sistema mantém um fila ativa, na qual todos os processos cujas imagens de memória estão em um repositório secundário ou em memória principal, prontos para serem executados. O tempo do procedimento de *swap* é extremamente alto e fica desabilitado até que muitos processos estejam executando e usando um valor limite de memória *RAM* [15].

## 2.7 Bibliotecas estáticas e dinâmicas

As bibliotecas estáticas são pacotes de coleções de arquivos de objeto e foram desenvolvidos com vistas ao reuso de código. Posteriormente, com o advento de sistemas operacionais multitarefa, outra forma de reuso se dá por meio de bibliotecas dinâmicas. Hoje, ambos os conceitos estão em uso, cada um com pontos fortes e fracos [16].

O conceito de bibliotecas estáticas consiste na tradução de arquivos fonte em objetos binários pelo compilador. Pode-se desejar manter os arquivos objetos para uso posterior, onde eles poderão ser facilmente combinados, em tempo de ligação, com arquivos de objetos de outros projetos. O ligador entende o formato de bibliotecas estáticas e é capaz de extrair seus conteúdos e fazer as ligações entre eles em tempo de compilação [16].

Em bibliotecas dinâmicas, é implementado um tipo de compartilhamento em tempo de execução. Na execução da aplicação, ela poderá incorporar partes já compiladas de outro executável, onde a integração poderá ocorrer sob demanda, em tempo de execução. Caso a aplicação necessite acessar alguma funcionalidade ausente, ele poderá simplesmente carregar uma biblioteca dinâmica [16].

A ligação com bibliotecas estáticas é uma boa escolha quando a aplicação acopla porções relativamente pequenas e em quantidades relativamente pequenas de bibliotecas estáticas. A ligação com bibliotecas dinâmicas se torna uma boa alternativa quando a aplicação depende de bibliotecas onde existe grande possibilidade de já estarem em execução [16].

Este capítulo abordou as principais ideias necessárias para que o projeto possa ser conduzido adequadamente, além de oferecer um suporte teórico para os procedimentos a serem realizados. O capítulo seguinte se dedicará a apresentar e detalhar os padrões internacionais de qualidade de imagens faciais que nortearam o desenvolvimento da ferramenta de análise de imagens a ser estudada e otimizada.

## Capítulo 3

# Padrões de qualidade para imagens faciais

No decorrer deste capítulo, serão abordadas questões relativas à qualidade de imagem para a utilização em documentos oficiais de viagem. A imagem é um componente importante e estabelecido pela principal agência de regulamentação da aviação civil (em nível global) como componente obrigatório em um documento de viagem válido. Essa agência definiu, por meio do documento nº 9.303, uma série de parâmetros de qualidade que uma imagem facial deverá cumprir, para que possa ser inserida em um documento oficial que identificará pessoas que transitarem entre diferentes países. Os padrões de qualidade de imagem facial, adotados no projeto **PhotoQuality**, seguem as normas contidas na regulamentação proposta pela referida agência, para que as imagens a serem validadas possam ser facilmente identificáveis por pessoas e/ou máquinas.

Os aspectos abordados neste capítulo tratarão principalmente dos requisitos e normas que são seguidos na avaliação da qualidade de imagens faciais. As normas e especificações citadas no decorrer do capítulo foram utilizadas pela equipe responsável como base para o desenvolvimento do subsistema **PhotoQuality**. Portanto, as seções seguintes não farão referência ao tema de avaliação de desempenho de sistemas computacionais. Esse tema voltará a ser abordado diretamente nos capítulos posteriores.

### 3.1 O Documento nº 9.303 *ICAO* de 2006

A Organização Internacional de Aviação Civil, do inglês *International Civil Aviation Organization (ICAO)*, consiste em uma agência ligada às Nações Unidas (ONU) para a regulamentação da aviação civil internacional. Ela é responsável por definir os princípios e as técnicas internacionais de voo. Além disso, trata da divulgação e do planejamento

para o desenvolvimento do transporte aéreo a nível global, com a intenção de assegurar o crescimento ordenado e a segurança da aviação civil mundial.

A ICAO é formada por um conselho que padroniza as práticas recomendadas para a infraestrutura de aviação, tráfego aéreo, inspeção aérea e prevenção contra possíveis falhas que possam pôr em risco a segurança dos usuários. A agência também regulamenta o formato de passaportes, ou documentos de viagem, passíveis de leitura por máquinas (*machine-readable passports*) mediante o documento nº 9.303, mantido pela organização [2].

As especificações contidas do documento nº 9.303 são publicadas em três partes distintas, uma para cada tipo de documento de viagem.

- Parte 1 – Especificações destinadas à emissão de passaportes.
- Parte 2 – Especificações destinadas à concessão de vistos.
- Parte 3 – Especificações destinadas à emissão de documentos oficiais de viagem (cartões de embarque).

Os três documentos estão divididos em uma estrutura em comum. As duas primeiras partes do documento tratam do material introdutório, definições e referências e a terceira parte contém especificações técnicas padronizadas para quaisquer documentos de viagens passíveis de serem lidos por máquinas ou *machine-readable travel documents (MRTDs)*. Os *MRTDs* possuem padrões de qualidade, formatação e requisitos físicos, tais como deformação, toxicidade, resistência a produtos químicos, estabilidade, umidade, luminosidade, além de incorporar mecanismos de segurança para evitar meios fraudulentos de utilização. Os *MRTDs* seguem um leiaute padrão para possibilitar facilidades na leitura, tanto visual quanto por máquina, das informações em qualquer um dos países que pertencem à ONU.

Embora o conteúdo das informações seja padronizado no que se refere à maneira de utilização, aos padrões de dimensão, à flexibilidade na inspeção visual e zonas de leitura por máquinas (dentro do documento de viagem), podem existir diferentes tipos de documentos, bem como também requisitos adicionais solicitados por algum estado ou organização. A terceira parte também trata de como o nome do proprietário deverá estar grafado em ambas as zonas e contém as especificações de como deverá ser a imagem de identificação.

As especificações das informações, que deverão ser únicas, são encontradas na parte 4, enquanto a parte 5 contém as especificações para o tamanho adicional em que o documento poderá ter variações. Vistos e cartões de embarque permitem dois tamanhos distintos, por isso cada uma das seções do documento contém uma 5ª parte para tratar sobre essas diferenças em cada um dos documentos.

A primeira parte do documento não conta com a seção dedicada à definição das múltiplas dimensões possíveis, já que passaportes têm apenas uma especificação de dimensões físicas. Na seção nº 4 da parte em questão, constam as especificações das dimensões únicas que devem ser seguidas, o leiaute das páginas com informações, o conteúdo, especificação de zonas de leitura, além do leiaute das informações dispostas em cada página, tais como especificações para a zona legível por máquinas ou *machine readable zone (MRZ)*. Demais requisitos incluem detalhes para a representação, dentro do passaporte legível por máquinas *MRP*, da organização ou país emissor, campo de nacionalidade e informações de datas, juntamente com as especificações referentes ao posicionamento da imagem facial e dígitos verificadores.

As especificações contidas no documento nº 9.303 permitem que documentos de viagem contenham mais de uma tecnologia de leitura disponível, o que inclui código de barra, tiras magnéticas, circuitos integrados ou mesmo memórias óticas coexistindo com tecnologias *OCR-B - Optical Character Recognition*. O modelo de reconhecimento de caracteres *OCR-B* foi aceito há mais de vinte anos com um método consistente para transcrição de documentos por favorecer a interoperabilidade entre vários equipamentos e seu custo razoavelmente inferior às demais alternativas.

### **3.1.1 Relevância da imagem facial para o e-passaporte**

A imagem facial é um componente essencial do passaporte, onde o agente poderá identificar de maneira preliminar o portador do passaporte, verificando se as características faciais estão de acordo com a imagem facial contida no documento e se o mesmo é válido para efeitos de identificação do portador. No momento da emissão do documento, a entidade responsável pela emissão do passaporte identifica o indivíduo e sua imagem facial. Caso haja diferenças de características faciais da foto, o documento não poderá ser emitido. Muitos países optam por gerar uma imagem facial no momento da emissão do passaporte, eliminando, assim, a possibilidade da imagem facial que estará no documento ser de outra pessoa.

Mesmo assim, há possibilidade de algum cidadão obter um passaporte válido de alguma nacionalidade, remover a imagem facial e substituí-la por uma que identifique outro portador, passando, assim, a ser identificado como outra pessoa. Nesse caso, trata-se de uma falsificação de documento que poderá trazer muitos problemas para os agentes de fronteira e, conseqüentemente, para a segurança do país. Esse fato levanta a necessidade de aperfeiçoar os métodos convencionais de identificação pessoal, mediante meios tecnológicos.

O passaporte biométrico vem de encontro à necessidade de criar meios que tornem complexo o processo de falsificação desses documentos, fazendo com que haja uma grande

redução nos índices desse tipo de falha de segurança. Assim, quando todos os países adotarem os padrões *ICAO* de identificação, espera-se que haja mais segurança nas fronteiras de todos os países membros.

A tecnologia presente nos e-passaportes consiste na armazenagem de todas as informações impressas no passaporte, juntamente com a imagem facial, em um *chip* que poderá ser lido por aparelhos de identificação. Os *chips* contêm informações que só podem ser acessadas por máquinas credenciadas, além de serem informações certificadas por meio de uma assinatura digital emitida por entidades idôneas. Caso haja alguma diferença dos dados impressos com os dados contidos no *chip*, ou os as informações contidas no chip não estão certificadas da maneira correta, o passaporte e o portador passam a ser suspeitos de fraude e ficam impedidos de adentrar no país, naquele momento.

A *ICAO* estabelece uma série de padrões para a imagem facial que irá ser utilizada no e-passaporte. Esses padrões tem o objetivo de estabelecer os requisitos básicos de como deverá ser o processo de captação da imagem, juntamente com os padrões de armazenamento no *chip* contido no passaporte.

## 3.2 Passaporte biométrico (e-passaporte)

O passaporte biométrico (digital), denominado e-passaporte ou ePassaporte, consiste em um documento de passaporte impresso em papel convencional, porém dotado de um *chip* capaz de armazenar informações de biometria. Método este que se revela muito eficiente na identificação de passageiros internacionais. O *chip* contém um microprocessador e mecanismos de comunicação embarcados e nos passaportes que seguem o padrão proposto pelo documento número 9303 da *ICAO*, poderá ser encontrado na capa frontal ou traseira, na parte central do passaporte.

Tecnologias biométricas têm mostrado ser a tendência nas metodologias de autenticação segurança e identificação pessoal. Uma grande quantidade de nações estão transferindo suas plataformas de identificação para um padrão que possibilite a leitura através de dispositivos eletrônicos, seguindo os padrões estabelecidos pela Organização de Aviação Civil Internacional (OACI). O conceito de *Machine Readable Zone (MRZ)* foi introduzido em 1980 com o conceito de conter um par de linhas legíveis por máquinas, na parte de baixo da página de identificação do passaporte. Entretanto, a OACI passou a adotar biometria em 1997 [2]. Com o aumento da rigidez na segurança contra fraudes, aumentou a necessidade de novas tecnologias capazes de prover sistemas de autenticação mais eficientes, em termos de autenticação pessoal. O padrão de biometria escolhido contém recursos biométricos, tais como impressão digital, reconhecimento facial e de íris e demais mecanismos de segurança.



O conceito de e-passaporte foi traduzido por Davida e Desmedt [17] [18] e estabelece que as informações contidas nos devem obedecer aos formatos recomendados e demais protocolos de comunicação estabelecidos no documento *ICAO* número 9303 [19].

### 3.2.1 Funcionamento do passaporte biométrico

Um e-passaporte é um documento básico de identificação e melhora os aspectos de segurança nos controles de fronteiras. Os escritórios de segurança de fronteira utilizam leitores para captar as informações dispostas nas MRZs e assim fazer as verificações cabíveis. As informações também poderão ser obtidas por meio da leitura do *chip*, ou *Radio Frequency Identification (RFID)*, embutido no documento de passaporte.

### 3.2.2 Propriedades particulares do e-passaporte

- Informações biográficas e biométricas são armazenadas digitalmente de maneira segura e que devem ser idênticas às que constam escritas [20].
- O *chip RFID* é o responsável por armazenar as informações.
- O *chip* contém uma assinatura digital para atestar sua autenticidade e a dos dados armazenados.

## 3.3 Padrões *ICAO* para fotos do e-passaporte

As especificações para a qualidade e confiabilidade da imagem constam no documento nº 9.303 do *ICAO* e se baseiam nos padrões e requisitos estabelecidos no ISO/IEC 19794-5, *Biometric Data Interchange Format – Face Image Data*, para o cumprimento dos parâmetros adequados de passaportes biométricos. As recomendações sobre o documento de viagem definem que os formatos padrão para imagens digitais e rostos devem assegurar que:

- Haja possibilidade de verificação visual da imagem, de modo que seja possível visualizar pequenos detalhes faciais, tais como pequenas cicatrizes e verrugas que poderão ser úteis na identificação facial.
- Haja possibilidade de verificação visual para efeitos de comparação entre uma pessoa presente e a imagem facial disposta no documento de identificação.
- Computadores possam efetuar identificação automatizada da face para a realização de pesquisa de dados em uma base de informações.

O padrão ISO/IEC 19794-5 também especifica requisitos para que as imagens possam ser lidas por qualquer aplicação executando em qualquer dispositivo, mesmo os que possuem poucos recursos em termos de hardware, de modo eficiente e preciso. Os padrões não dizem respeito apenas ao formato da imagem, mas abrangem também aspectos como:

- Questões de enquadramento (pose, expressão facial e etc.);
- Propriedades fotográficas (luminosidade, posicionamento, foco e etc.);
- Atributos de imagem digital (resolução da imagem, tamanho etc...).

O *ICAO* publicou no documento nº 9.303, parte I, volume I alguns exemplos de imagens que são aceitas ou não para serem colocadas em um documento de passaporte. Vários países utilizam as recomendações da *ICAO* também para imagens que constam em documentos de identidade ou carteiras de habilitação.

Observações: As imagens utilizadas foram retiradas dos exemplos contidos no documento nº 9.303 do *ICAO*.

As especificações de qualidade da imagem são referentes aos seguintes aspectos:

- Posicionamento/enquadramento do sujeito e fundo;
- Pose;
- Qualidade da imagem;
- Cores e luminosidade;
- Penteado do sujeito e expressão facial;
- Luminosidade aplicada na face;
- Óculos;
- Chapelaria;
- Crianças.

### **3.3.1 Posicionamento/enquadramento do sujeito**

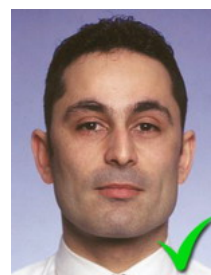
O enquadramento do sujeito, em relação à borda da imagem, não deverá ser excessivamente próximo ou muito longe, de maneira que a distância entre os olhos não deverá ser maior nem menor que um quarto da largura que a imagem possui. O rosto deverá estar centralizado tanto verticalmente quanto horizontalmente. O fundo deverá ter um único padrão de cor que não comprometa a imagem do sujeito. A figura 3.1 ilustra esta especificação.



Enquadramento  
muito próximo



Enquadramento  
muito distante



Enquadramento  
correto



Fundo poluído



Face descentrali-  
zada

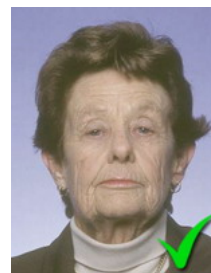


Imagem facial  
aceita

Figura 3.1: Especificação para enquadramento e posicionamento de indivíduos em imagens contidas em documentos de viagens [2]

### 3.3.2 Pose

O sujeito deverá estar olhando para a câmera frontalmente, sem exibir um ombro mais alto que o outro. A posição da cabeça deverá estar completamente na vertical, de modo que, traçando-se uma linha imaginária na altura dos olhos, ela esteja paralela à borda superior da imagem. A figura 3.2 ilustra esta especificação.



Elevação de om-  
bro



Olhar lateral

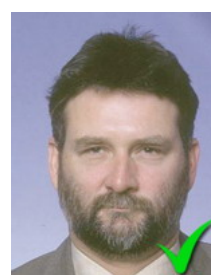


Imagem facial  
correta

Figura 3.2: Especificação da pose do indivíduo em imagens contidas nos documentos de viagem [2]

### 3.3.3 Qualidade da imagem facial

O retrato deverá estar ajustado para partir do nariz para as orelhas, sem a presença de manchas na imagem ou qualquer marca que não faça parte do rosto do sujeito. Os tons de pele deverão estar naturais. A figura 3.3 ilustra esta especificação.

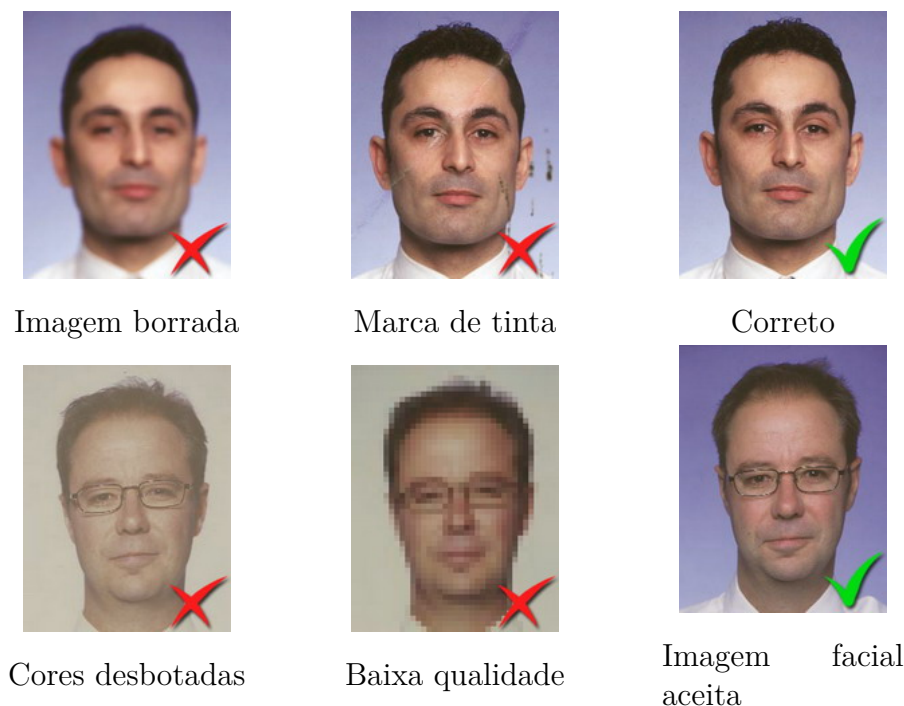


Figura 3.3: Especificação de qualidade de imagens contidas nos documentos de viagem [2]

### 3.3.4 Esquema de cores e iluminação

O retrato deverá apresentar cores neutras em que o sujeito está de olhos abertos e visíveis. É recomendado um balanço de exposição entre branco e cinza de 18%. Recomenda-se no mínimo 128 tonalidades de cores na área do rosto. O padrão ISO/IEC não permite que haja alteração digital da imagem para correção do esquema de cores e demais fatores.

### 3.3.5 Penteado e expressão facial

O penteado não deverá estar cobrindo áreas importantes da face, tais como sobrancelhas e olhos. O sujeito deverá estar olhando frontalmente a câmera com expressão neutra e não estar de boca aberta. A figura 3.4 ilustra esta especificação.

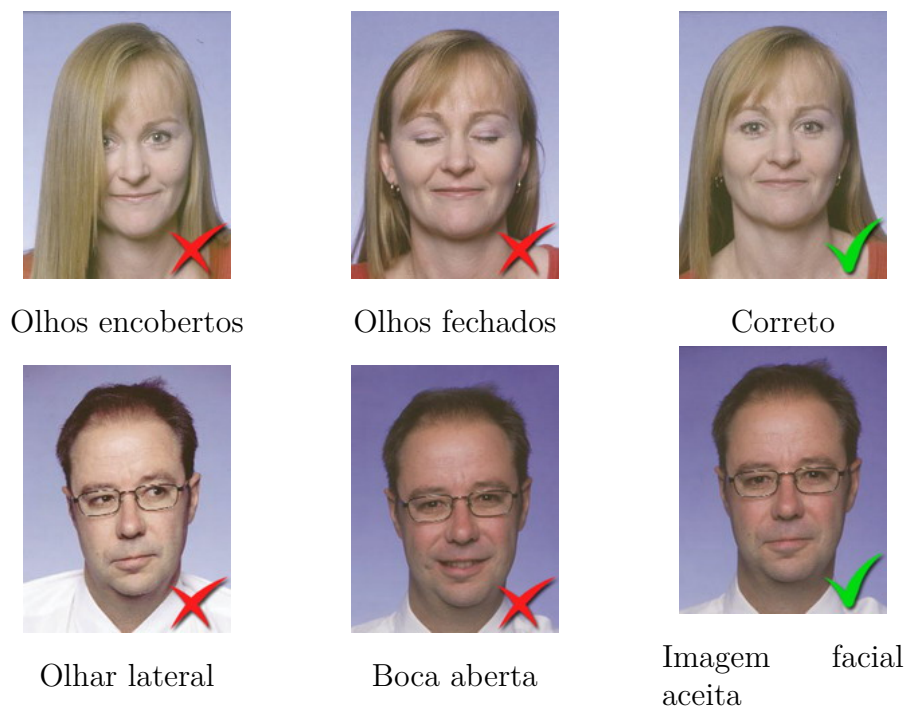


Figura 3.4: Especificação de penteado e expressão facial do indivíduo em imagens contidas nos documentos de viagem [2]

### 3.3.6 Luminosidade e fundo

A luminosidade deverá estar uniforme, sem a presença de sobressalientes ou reflexão de luz pelo rosto do sujeito, sem olhos vermelhos e o fundo não poderá conter sobreposição visível. A figura 3.5 ilustra esta especificação.

### 3.3.7 Óculos

O retrato deverá exibir qualquer reflexo produzido pelas lentes dos olhos. Estas lentes deverão ser transparentes. As armações não poderão ser de material espesso e fosco, além de não poder encobrir partes dos olhos. A figura 3.6 ilustra esta especificação.

### 3.3.8 Chapelaria

As chapelarias não são aceitas, exceto em circunstâncias em que o Estado emissor permite a utilização. As circunstâncias poderão ser de caráter religioso, médico ou cultural. A figura 3.7 ilustra esta especificação.



Face refletindo luz



Olhos vermelhos



Correto



Sombra visível no fundo



Sombra na face



Imagem facial aceita

Figura 3.5: Especificação de luminosidade e plano de fundo das imagens contidas nos documentos de viagem [2]



Óculos escuros



Reflexo nas lentes



Correto



Armações espessas



Armação encobrindo os olhos

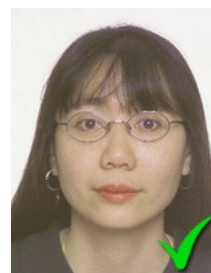


Imagem facial aceita

Figura 3.6: Especificação para a utilização de óculos por indivíduos em imagens contidas nos documentos de viagem [2]



Chapelaria não permitida



Chapelaria não permitida



Correto



Face encoberta



Sombras ofuscando o rosto

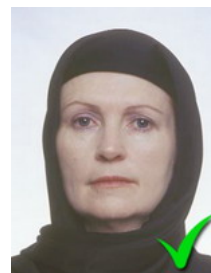


Imagem facial aceita

Figura 3.7: Especificação para a utilização de chapelaria por indivíduos em imagens contidas nos documentos de viagem [2]

### 3.3.9 Especificação para crianças

Em alguns casos, expressões faciais não neutras e olhos fechados são aceitáveis. Porém, a criança deverá estar sempre sozinha, sem encostos de cadeira ou brinquedos em vista. A figura 3.8 ilustra esta especificação.



Presença de outra pessoa



Presença de brinquedos



Imagem facial aceita

Figura 3.8: Especificação captura de imagens faciais de crianças [2]

Este capítulo abordou os aspectos relativos aos padrões internacionais para a análise de imagens faciais. Foi dado um pequeno resumo do documento *ICAO* nº 9.303, além

de uma introdução ao funcionamento dos passaportes biométricos e as especificações de qualidade de imagens faciais que estarão contidas nestes documentos. O próximo capítulo tem como foco o detalhamento do sistema de análise das imagens e os demais subsistemas que compõem a plataforma.



## Capítulo 4

# Sistema de análise de imagens faciais

O sistema consiste em uma plataforma destinada exclusivamente para fornecer o serviço de análise e validação de imagens. É acessível por meio de uma rede local, onde todos os demais sistemas que estão conectados podem consumir o serviço. O sistema é compatível com os demais sistemas, independentemente da plataforma de execução. Os únicos requisitos necessários para consumir o serviço são: possuir as interfaces de acesso à rede *LAN* através dos protocolos *TCP/IP* e suportar a decodificação da linguagem *Web Services Description Language (WSDL)*.

A plataforma é composta por dois subsistemas principais. O primeiro é denominado **PhotoQuality**, cuja função é a análise das imagens faciais propriamente ditas. O segundo subsistema foi denominado **PSI** e tem como função prover uma interface de acesso através da rede, gerenciar múltiplas requisições ao sistema e orquestrar a chamada ao subsistema **PhotoQuality**. Esses dois subsistemas estarão implantados em uma única máquina.

### 4.1 O subsistema **PhotoQuality**

O **PhotoQuality** é produto de um projeto desenvolvido pelo Centro de Pesquisa em Arquitetura da Informação (CPAI/UnB), que pretende auxiliar nas verificações de autenticidade dos candidatos que fazem provas de concursos, vestibulares e avaliações, conduzidas pelo Cespe-Cebraspe. O emprego de técnicas de processamento de imagens e visão computacional pode ser de grande utilidade na verificação de autenticidade. Dentre as possibilidades de aplicação da análise automática da qualidade de imagem, independentemente do dispositivo de aquisição, tem chamado a atenção de algumas empresas. O desafio está em padronizar a imagem obtida, focando-se exclusivamente no resultado final da aplicação. O objetivo do projeto é oferecer uma ferramenta que seja capaz de avaliar uma determinada imagem facial de uma pessoa de maneira automática, a partir de um conjunto de requi-

sitos e métricas. O **PhotoQuality** foi desenvolvido para seguir as normas internacionais contidas no documento nº 9.303 da *ICAO*.

Um operador humano responsável por analisar a qualidade das imagens submetidas é inviável, pois, além de tornar o processo ineficiente - devido ao caráter manual -, poderá trazer uma série de problemas de consistência no processo, uma vez que a avaliação possui caráter subjetivo. A automatização do procedimento introduzirá melhorias significativas no desempenho e na confiabilidade da avaliação.

#### **4.1.1 Finalidade do projeto**

O projeto tem como finalidade desenvolver uma implementação que seja capaz de realizar a análise e avaliação da qualidade de imagem de forma automática a partir de requisitos de qualidades previamente definidos que permitam a correta detecção do indivíduo, por operador humano, com o máximo de precisão possível.

#### **4.1.2 Processo de análise automática de imagens**

As aplicações em Processamento Digital de Imagens podem ser classificadas em baixo, médio e alto nível. O baixo nível é definido por operações de imagem em cores para níveis de cinza, entre outros, sendo que a entrada e saída do processo é uma imagem. Aplicações de médio nível utilizam imagem como entrada e na saída são obtidas as bordas, contornos, quantidades de objetos, tamanho de objetos e demais características dessas imagens. Aplicações de alto nível têm como objetivo encontrar sentido na imagem, de modo a detectar se há representação de algum objeto conhecido, o que implica a utilização de funções cognitivas relativas à visão. Aplicações de alto nível pertencem à área de Visão Computacional [21].

#### **4.1.3 Análise da qualidade de imagens digitais**

A qualidade de uma imagem digital poderá ser afetada por uma série de fatores, tais como o conjunto de sensores utilizados no momento da captura, conjunto ótico, o ambiente e o tipo de processamento utilizado. Esses fatores evidenciam alguns parâmetros:

1. Resolução espacial – Menor detalhe discernível em uma imagem [21];
2. Contraste – Diferença entre a intensidade máxima e mínima do pixel em uma imagem;
3. Ruído – Processo que afeta a imagem resultante e não faz parte do cenário (sinal obtido inicialmente) [21].

Métodos de redução do nível de ruído, por meio de algoritmos, têm sido estudados e implementados pela indústria.

#### 4.1.4 Detecção de Indivíduos por imagens

O grande desafio para sistemas que efetuam avaliação de qualidade de imagens automaticamente consiste na ausência de padronização das imagens oferecidas pelos indivíduos ou dos métodos de captura. Os problemas podem ser baixa resolução espacial ou excesso de ruído que dificulta o reconhecimento, o que compromete a consistência em termos de garantia de unicidade.

Para resolver esse problema, a adoção de um padrão de qualidade de imagens com a finalidade de reconhecimento individual é relevante. As iniciativas para padronizar imagens de indivíduos são mantidas principalmente por órgãos de segurança pública, mas sempre com base em normas internacionais já estabelecidas, como é o caso dos padrões estabelecidos pelo documento *ICAO* n° 9.303.

#### 4.1.5 Desenvolvimento

O processo de desenvolvimento foi dividido em cinco etapas diferentes, que vão desde a definição de conceitos que nortearão o processo até o desenvolvimento de um teste de *benchmark* para facilitar os ajustes pendentes. Essas etapas contêm as atividades respectivas a serem realizadas para que cada etapa seja concluída. Na tabela 4.1, encontram-se as etapas e as atividades referentes a cada uma delas.

#### 4.1.6 Responsáveis pelo projeto

O projeto é conduzido por uma equipe composta por quatro pesquisadores, sendo que três deles são seniores e dois juniores. Todos pertencentes à equipe de Processamento de Sinais do CPAI/UnB. Os pesquisadores são:

##### Pesquisadores Seniores

- Prof. Dr. Flávio de Barros Vidal (Coordenador);
- Prof. Dr. Alexandre Zaghetto;
- Prof. Dr. Bruno Luigi Macchiavello.

##### 4.1.4.2 Pesquisadores Juniores

- Luan Caius Ramos;
- Luiz Henrique Aguiar.

Tabela 4.1: As atividades envolvidas em cada etapa do projeto

Etapa	Atividade(s)
Definição do Protocolo Simples de Avaliação das imagens	<ul style="list-style-type: none"> <li>• Estudo das Normas Internacionais sobre qualidade de imagens faciais digitais.</li> <li>• Levantamento bibliográfico dos principais trabalhos acadêmicos realizados sobre o tema.</li> </ul>
Implementação do Protocolo Simples	<ul style="list-style-type: none"> <li>• Desenvolvimento protótipo inicial <i>standalone</i>.</li> <li>• Testes em um conjunto reduzido de imagens faciais digitais, disponibilizadas pelo Cespe-Cebraspe, categorizadas de acordo com os protocolos definidos na etapa anterior.</li> </ul>
Aplicação do algoritmo desenvolvido nas imagens de Testes	<ul style="list-style-type: none"> <li>• Realização de testes utilizando o conjunto ampliado de imagens faciais para determinação do grau de desempenho atingido no protótipo inicial.</li> <li>• Definir novos ajustes e métricas para melhorar dos resultados obtidos no grupo inicial de testes.</li> </ul>
Criação do <i>Benchmark</i> de imagens para Avaliação	<ul style="list-style-type: none"> <li>• Desenvolvimento do <i>benchmark</i></li> </ul>
Refinamento do algoritmo desenvolvido a partir do <i>benchmark</i>	<ul style="list-style-type: none"> <li>• Realização dos ajustes finos.</li> <li>• Validação dos ajustes novamente pelo <i>benchmark</i>.</li> </ul>

### 4.1.7 Características e Funcionalidades

O acesso às funcionalidades oferecidas pela ferramenta é possível somente dentro do dispositivo que possui o executável e suas dependências. A chamada ao processo é feita via *Command-line Interface* do *Windows*, passando como argumentos o modo de análise desejado, juntamente com o endereço da imagem que se deseja analisar. O caminho da imagem poderá apontar para um local dentro do disco local da máquina, para um endereço em rede local ou ainda para uma *Unified Resource Locator (URL)*. O retorno é uma *string* com cada um dos parâmetros de qualidade analisados.

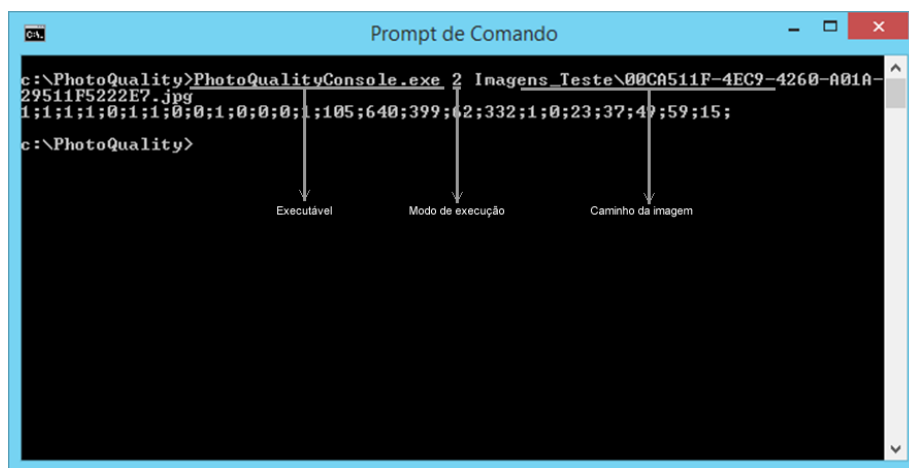


Figura 4.1: O processo de chamada do subsistema PhotoQuality

O subsistema *PhotoQuality* não foi projetado inicialmente com um componente nativo capaz de prover interface com redes locais. Assim, sem o auxílio de outro subsistema, ele é incapaz de receber requisições através de uma rede *Local Area Network (LAN)*, por meio de protocolos *TCP/IP*, sendo, a princípio, inadequado para trabalhar de modo *on-demand*, onde as requisições deverão ocorrer de maneira assíncrona e simultânea por múltiplos sistemas em dispositivos diferentes.

O sistema foi desenvolvido de maneira modular. Há um módulo principal e demais módulos, que são responsáveis por verificar uma imagem em cada uma das especificações de qualidade. O módulo principal instancia os demais, que são executados de maneira sequencial. O resultados retornados por cada um dos módulos de análise são acumulados internamente pelo módulo principal. Para fins de eficiência, caso o resultado de um módulo anterior não seja minimamente tangível, do ponto de vista do resultado esperado, a execução dos próximos a serem executados poderá ser abortada.

A maioria das especificações de qualidade implementadas nos módulos têm base nos padrões definidos no documento *ICAO* nº 9.303. Há também algumas especificações de avaliação que foram implementadas com base em requisitos demandados pelo patrocinador do projeto.

Cada especificação possui um par de parâmetros que prescrevem os valores mínimos e máximos para que uma imagem seja aceita. Os valores dos parâmetros são armazenados em arquivo de configuração. Esses valores foram previamente acertados por definição do patrocinador ou foram obtidos mediante calibragem onde um grupo de avaliadores analisou um grande conjunto de imagens, com base nas especificações de qualidade. Os valores dos parâmetros poderão ser alterados a critério do proprietário do sistema.

Os módulos do PhotoQuality que avaliam a qualidade da imagem são:

1. **Verificador de formato:** Este módulo atende a uma especificação demandada pelo patrocinador. Avalia se o formato do arquivo de imagem digital enviado está dentro dos padrões estabelecidos. Este módulo avalia se as imagens estão nos formatos *bmp* e *jpg*.
2. **Avaliador do tamanho da imagem:** Este módulo foi requisitado pelo patrocinador. Informa o tamanho de imagem em *Kilobytes*. Os valores mínimos e máximos para o parâmetro de avaliação foram definidos de modo arbitrário.
3. **Verificador de altura:** Módulo requisitado pelo patrocinador. Avalia a altura da imagem em *pixels*. Os valores foram definidos arbitrariamente.
4. **Avaliador de largura:** Módulo requisitado pelo patrocinador. Avalia a largura da imagem em *pixels*. Os valores foram definidos arbitrariamente.
5. **Verificador de razão:** Módulo prescrito pelo patrocinador. Avalia se as proporções entre a largura e a altura da imagem estão dentro dos valores definidos. Esses valores foram fixados arbitrariamente pelo patrocinador.
6. **Avaliador dos níveis de borramento:** Módulo implementado com base nas especificações contidas no capítulo 3, na seção 3.3.4. Os valores mínimos e máximos aceitos para esta especificação foram obtidos mediante calibragem.
7. **Medidor da quantidade de faces:** Módulo implementado com base nas especificações contidas no capítulo 3, na seção 3.3.8 e na seção 3.3.9. Os valores mínimos e máximos aceitos para esta especificação foram obtidos mediante calibragem.
8. **Verificador de pares de olhos:** Módulo implementado com base nas especificações contidas no capítulo 3, na seção 3.3.5 e na seção 3.3.7. Os valores mínimos e máximos aceitos por este módulo foram obtidos mediante calibragem.
9. **Verificador de altura da face:** Módulo implementado com base nas especificações contidas no capítulo 3, na seção 3.3.1. Os valores mínimos e máximos aceitos para esta especificação foram obtidos mediante calibragem.

10. **Verificador de largura da face:** Módulo que implementado com base nas especificações contidas no capítulo 3, na seção 3.3.1. Os valores mínimos e máximos aceitos para esta especificação foram obtidos mediante calibragem.
11. **Verificador do posicionamento da face:** Módulo desenvolvido com base nas especificações contidas no capítulo 3, na seção 3.3.2. Os valores mínimos e máximos aceitos para esta especificação foram obtidos mediante calibragem.
12. **Verificador de artefatos e ruídos:** Módulo que verifica, no plano de fundo da imagem, a existência de artefatos, medindo também os níveis de ruído desta região. Elaborado com base nas especificações contidas no capítulo 3, na seção 3.3.3 e na seção 3.3.6. Os valores mínimos e máximos aceitos por este módulo foram obtidos mediante calibragem.
13. **Verificador dos níveis de contraste:** Módulo que avalia os níveis de contraste da face em relação ao fundo, com base nas especificações contidas no capítulo 3, na seção 3.3.4 e na seção 3.3.6. Os valores mínimos e máximos aceitos por este módulo foram obtidos mediante calibragem.

## 4.2 O subsistema PSI

O subsistema PSI consiste em um serviço da plataforma **Microsoft Windows®**. Serviço é um tipo de aplicação que é executada em segundo plano pelo sistema. Desse modo, há um processo permanentemente ativo no servidor atendendo requisições de análise. Por meio desse serviço, é possível oferecer as funcionalidades adicionais que são necessárias para o sistema, tais como interface para acesso através da rede local, gerenciamento das requisições e formatação das respostas.

O serviço foi desenvolvido utilizando o *framework* para aplicações distribuídas da Microsoft, denominado *Windows Communication Foundation (WCF)*. O *WCF* é um *framework* para o desenvolvimento de aplicações orientadas a serviço. Usando *WCF*, é possível enviar dados como mensagens assíncronas de um nó de extremidade para outro. Um nó pode ser parte de um serviço disponível continuamente hospedado em um servidor *Web Information Services (IIS)* ou poderá ser um serviço hospedado em uma aplicação, ser um serviço ou mesmo um cliente que requisita dados de outro. A mensagem pode ser simples, contendo apenas cadeias de caracteres, ou complexa, como um *stream* ou dados binários [22].

### 4.2.1 Módulos

Para atender às funcionalidades necessárias, o subsistema foi desenvolvido em módulos cujo papel é atender cada um dos requisitos necessários. A estratégia modular também objetiva facilitar as manutenções futuras a que o subsistema poderá ser submetido. Os módulos do PSI são:

- **Gerenciador de requisições:** Este módulo é responsável por receber as requisições e encaminhá-las para processamento, recursos disponíveis. Em situações em que o hardware já esteja processando outras requisições, este módulo também trata de manter uma fila para inserir requisições ainda não atendidas. Assim, o módulo gerencia quais deverão ser processadas e em qual momento. Este módulo é implementado pelo *WCF* e foi incorporado ao PSI.
- **Arquivos:** Este módulo é responsável por converter os arquivos de imagem para formatos adequados, providenciar o armazenamento das imagens no disco local do servidor e a exclusão posterior destes arquivos, após análise.
- **Formatador de respostas:** Este módulo trata da formatação das respostas geradas pelo subsistema *PhotoQuality* para um padrão que possa ser facilmente decodificado pelos nós de extremidade cliente.
- **Orquestrador:** Módulo responsável pela formatação das requisições, chamada dos módulos de arquivos e formatação de respostas, além da chamada ao subsistema *PhotoQuality*.

Assim que o módulo gerenciador de requisições recebe uma nova requisição com uma imagem para ser analisada, ele verifica se há recursos físicos para atender à requisição. Caso haja requisições pendentes, elas serão encaminhadas ao módulo orquestrador que faz o trabalho de formatação, chamada de módulo de arquivos, o qual nomeia a imagem e salva no disco. Em seguida, chama o subsistema *PhotoQuality* com os parâmetros corretos. Ao receber o resultado da análise, o orquestrador encaminha para o próximo módulo, que formatará a resposta. Em seguida, a resposta é encaminhada ao módulo gerenciador de requisições para ser enviada ao cliente que solicitou. O processo é ilustrado no fluxograma contido na Figura 4.2.

### 4.2.2 Características

As principais características do PSI são:

- Serviço independente que recebe requisições de análise de imagens, formata a requisição e repassa para o *PhotoQuality*.



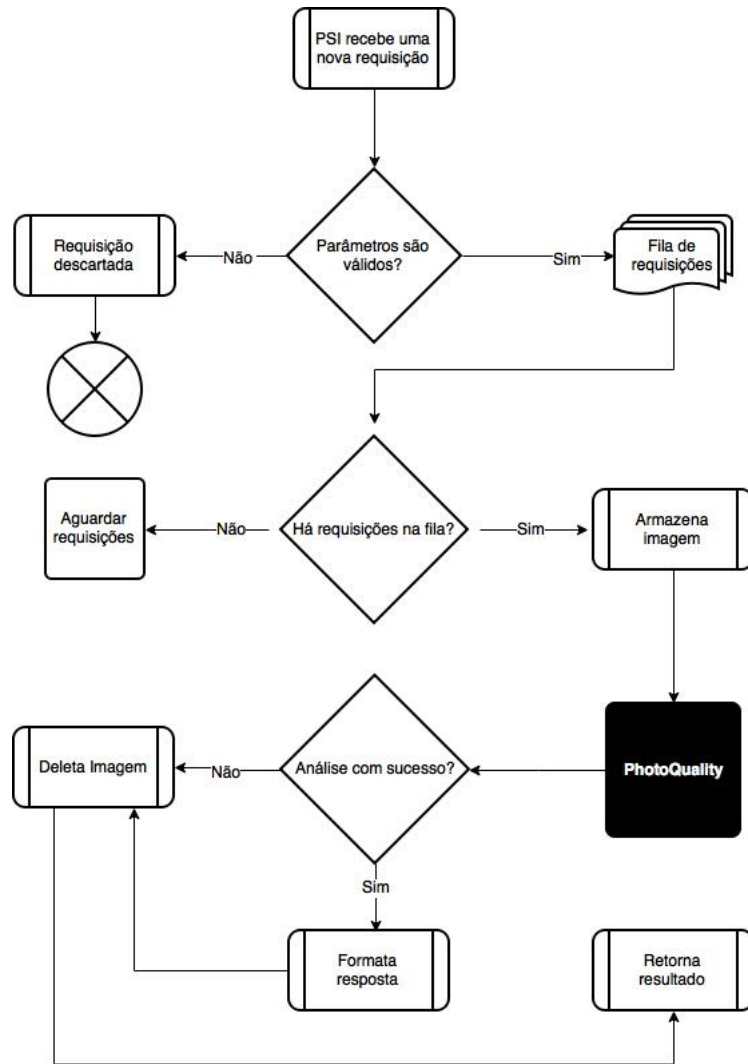


Figura 4.2: Fluxograma de funcionamento do subsistema PSI

- Ao receber o resultado do processamento o serviço, a resposta é formatada e enviada.
- Desenvolvido em *c# .NET* utilizando o *framework WCF*.
- Recebe como argumento a imagem no formato de um *array* de *bytes*.
- O serviço PSI funciona exclusivamente sobre o protocolo *TCP/IP*.
- Aceita requisições através da porta *TCP 8523*.

### 4.3 Infraestrutura

É de grande importância saber as configurações do ambiente em que a plataforma de análise de imagens está. No âmbito de servidores físicos, é indispensável fazer o levantamento

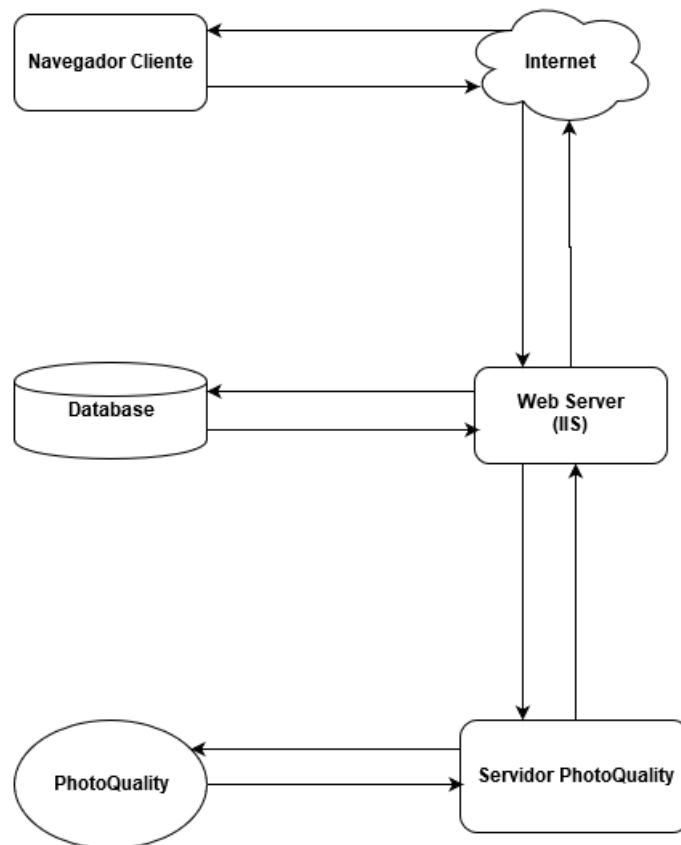


Figura 4.3: Processo de análise de uma imagem

prévio dos componentes físicos e suas capacidades. Por outro lado, o sistema operacional também é um fator importante, pois é a camada de *software* que permite acesso e gerência a esses recursos físicos.

A equipe responsável pela análise e otimização, apesar de poder acessar as configurações do ambiente de produção e coletar informações sobre os dispositivos, não poderá efetuar os testes de desempenho e coleta de dados diretamente nesse ambiente por determinação do setor de infraestrutura da instituição. Os testes de desempenho e coleta de resultados deverão ser executados em um ambiente propício à realização ao desenvolvimento. O hardware contido nesse ambiente possui, em geral, as mesmas configurações de *hardware* e *software* do ambiente de produção. A escolha por um ambiente mais idêntico o possível ao que irá executar a plataforma em definitivo tem o objetivo de tornar os resultados mais próximos aos que seriam obtidos no ambiente de produção. As configurações dos dispositivos de *hardware* do ambiente de desenvolvimento e produção são:

- Processadores: Intel Xeon E5-2260 2.20 GHz – *Octa-core* + *HyperThreading*
- Memória RAM: 14 GB
- Disco Rígido: 350 GB.

Com base em informações preliminares repassadas pela equipe de infraestrutura, foi possível saber que a rede que conecta os servidores conta com uma capacidade de banda de 1 Gbps para a rede *LAN* que conecta os servidores.

A instituição utiliza servidores virtualizados executando o sistema operacional da *Microsoft* como padrão. Assim, os servidores Web e de análise de imagem são todos *Windows*, sendo que a versão do servidor de análise de imagens é a 2012 R2 e a versão do sistema operacional servidor Web é 2008 R2. O servidor Web em si se trata do Internet Information Services (*IIS*) v8.

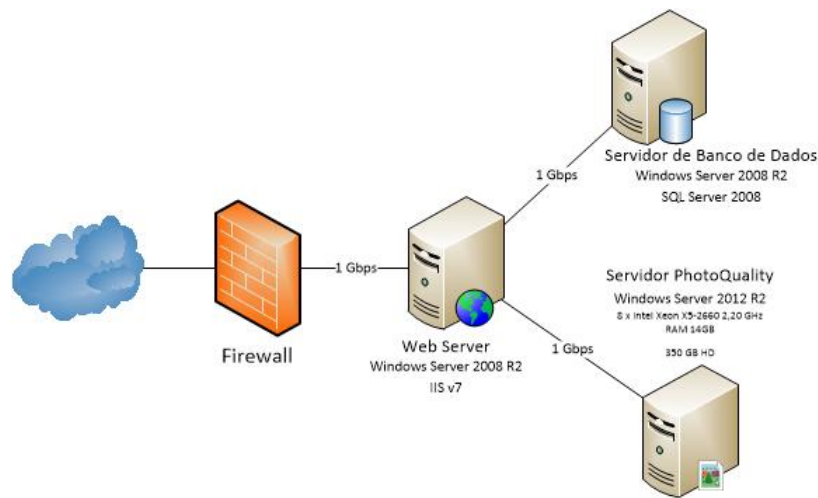


Figura 4.4: Diagrama da infraestrutura de execução da plataforma

Este capítulo objetivou dar uma visão panorâmica do sistema de análise de imagens faciais. Foram abordados cada um dos subsistemas que constituem a plataforma: o **PhotoQuality**, que realiza o trabalho de análise de imagens faciais, e o **PSI**, cuja função é prover interface de rede, gerenciamento das requisições e demais tarefas já citadas. Os módulos que desempenham cada uma das funções foram listados e explanados. O capítulo seguinte buscará levantar e validar as hipóteses para o baixo desempenho observado no sistema. Além disso, o capítulo também apresentará as alternativas disponíveis para a otimização do sistema, levando em consideração as hipóteses anteriormente levantadas.

## Capítulo 5

# Avaliação e estratégias para a otimização do desempenho

A formulação das hipóteses para o desempenho insatisfatório do sistema exige uma análise preliminar, mediante a execução de testes de desempenho sob alta taxa de utilização. Nessa etapa, as observações não serão focadas nas métricas de desempenho em si, mas nos índices de utilização de cada um dos dispositivos utilizados durante a execução do teste. Por esse motivo, as observações não estarão dentro de um contexto de cenários de desempenho, servirão de base para traçar as possíveis causas da limitação de desempenho observada e fundamentar os aspectos que são passíveis de otimização.

O processo de observação das taxas de utilização dos dispositivos pode ser útil para nortear as modificações que deverão ser feitas no modelo de execução ou mesmo em nível de código para oferecer um desempenho satisfatório. Espera-se com essas observações detectar componentes físicos que podem estar sobrecarregados e limitando o desempenho dos demais. Há também a possibilidade de haver componentes que estão sendo subutilizados e que podem ter sua carga de trabalho aumentada. Todas essas possibilidades citadas serão verificadas no decorrer deste capítulo.

Os testes de desempenho não poderão ser efetuados no ambiente de produção, pois estão inacessíveis para a realização de testes de desempenho. No momento do início do projeto de avaliação do desempenho, o sistema já está sendo utilizado em produção e a instalação de ferramentas de teste poderá impactar significativamente o ambiente. Utilizar um ambiente simulado para possibilita gerar resultados próximos aos que seriam obtidos no ambiente de produção, desde que a configuração de *hardware* e *software* sejam semelhante ao ambiente principal de execução, como é o caso.

## 5.1 Avaliação da utilização dos componentes físicos

Avaliar a utilização dos componentes físicos é uma etapa que objetiva verificar a hipótese de que a sobrecarga de utilização de alguns dos componentes de *hardware*, durante a análise das imagens, possa ser a causa do desempenho observado. Outro ponto importante é ter a noção de quais dispositivos podem estar ociosos. E, caso se julgue oportuno, alocar mais trabalho a esse componente, de modo que se traduza em ganhos no desempenho.

A avaliação dos recursos de hardware, realizada nesta etapa, diz respeito aos processos do subsistema **PhotoQuality**. Desse modo, os demais sistemas em execução não terão influência nas informações apresentadas, evitando, assim, a interferência desses sistemas nos resultados. Os dispositivos a serem monitorados, durante a execução da análise das imagens, foram a *CPU*, os dispositivos de entrada e saída (*E/S*) e a memória *RAM*.

No decorrer da avaliação da utilização de componentes físicos, a ser realizada no âmbito deste capítulo, será utilizado o *software* de monitoramento do desempenho padrão de sistemas operacionais *Windows*, denominado *Performance Monitor (Perfmon)*. Este monitor está presente gratuitamente em qualquer versão recente do sistema operacional, seja ela voltada para servidores ou computadores pessoais.

Esse monitor pode ser utilizado para examinar, em tempo real, como os programas em execução afetam o desempenho do computador, coletando dados de *log* para análise posterior. Utiliza contadores de desempenho, dados de rastreamento de evento e informações de configuração, que podem ser combinados nos Conjuntos de Coletores de Dados. Os contadores de desempenho são medidas do estado ou atividade do sistema. Podem ser incluídos no sistema operacional ou fazer parte de aplicativos individuais. O *Perfmon* solicita o valor atual dos contadores de desempenho em intervalos de tempo especificados [23].

Neste trabalho, o monitor é configurado para trabalhar *online*, onde os estados são capturados e armazenados com uma certa frequência predeterminada. O monitor captura os dados dos processos do **PhotoQuality** através dos contadores listados na Tabela 5.1, a cada um segundo. Como esse é um monitor executado na máquina que hospeda o serviço, além de ser configurado para uma resolução relativamente alta, existe um certo *overhead* relacionado.

Os resultados foram obtidos através de dez observações. Cada uma dessas observações foi realizada utilizando (como carga de trabalho) imagens selecionadas aleatoriamente, a partir de um conjunto de 39 mil imagens. O valor de cada observação consiste na média aritmética dos resultados de cada um dos contadores, após a execução da análise de duas mil imagens. Os contadores, suas definições, bem como os dispositivos a que avaliam, podem ser verificados na Tabela 5.1.

Tabela 5.1: Relação entre dispositivos avaliados e os contadores utilizados com suas definições.

Dispositivo(s)	Contador	Definição
Processador	% tempo de processador	Porcentagem de tempo decorrido em que todos as <i>threads</i> do processo usaram o processador para executar instruções. Uma instrução é a unidade básica de execução em um computador, uma <i>thread</i> é o objeto que executa instruções e um processo é o objeto criado quando um programa é executado. O código executado para lidar com algumas interrupções de hardware e condições de desvio é incluído nessa contagem.
Memória	<i>Bytes</i> particulares	Tamanho atual, em <i>bytes</i> , da memória alocada pelo processo que não pode ser compartilhada com outros processos.
E/S	<i>Bytes</i> de E/S	<ul style="list-style-type: none"> <li>• <b>Leitura:</b> A taxa a qual o processo está lendo <i>bytes</i> de operações de E/S. Esse contador contabiliza toda a atividade de E/S gerada pelo processo, inclusive E/S de arquivo, rede e dispositivo.</li> <li>• <b>Escrita:</b> A taxa a qual o processo está gravando <i>bytes</i> para operações de E/S. Esse contador contabiliza toda a atividade de E/S gerada pelo processo, inclusive E/S de arquivo, rede e dispositivo.</li> </ul>
	Operações de E/S	<ul style="list-style-type: none"> <li>• <b>Leitura:</b> A frequência com a qual o processo está gerando operações de E/S de leitura. Esse contador considera toda a atividade de E/S gerada pelo processo, incluindo E/S de arquivo, rede e dispositivo.</li> <li>• <b>Escrita:</b> A frequência com a qual o processo está gerando operações de E/S de gravação. Esse contador considera toda a atividade de E/S gerada pelo processo, incluindo E/Ss de arquivo, rede e dispositivo.</li> </ul>

Para garantir a mínima interferência de outros processos em execução, a prioridade do sistema em teste foi ajustada para alta. Deste modo, o sistema operacional concederá, em geral, maiores intervalos de tempo para o processo do **PhotoQuality** e suas *threads* permanecerem em execução no processador, de modo que os demais processos em execução interfiram minimamente no desempenho observado.

### 5.1.1 Utilização dos dispositivos de E/S

A observação do acesso aos recursos de entrada e saída pelo sistema **PhotoQuality** é importante para observar a frequência e proporção com que o sistema de análise de fotografias está acessando esses dispositivos. Acessos aos dispositivos de E/S costumam demandar um tempo elevado, se comparados aos tempos demandados pelos demais componentes. Assim, a hipótese de que o desempenho insuficiente é decorrente do acesso inadequado aos dispositivos de entrada e saída deve ser verificada.

Para ter uma dimensão do quanto os dispositivos de entrada e saída estão sendo demandados pelo **PhotoQuality**, foram monitorados dois contabilizadores de desempenho. O primeiro consiste no volume de dados transmitidos por segundo, a partir dos dispositivos de E/S existentes. O outro contabilizador permitiu obter a quantidade de operações de leitura e escrita realizadas a cada segundo.

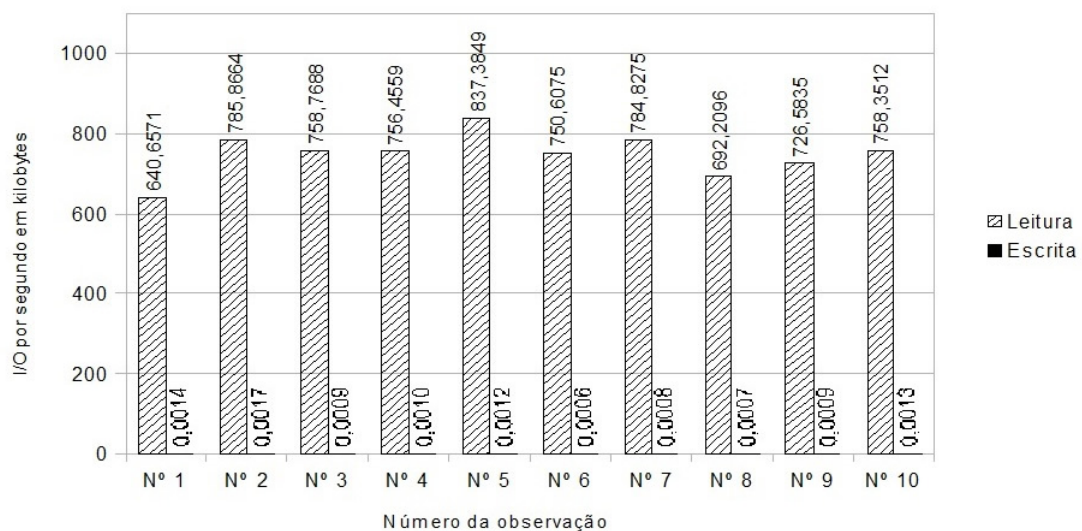


Figura 5.1: Volume de transferência de dados dos dispositivos de entrada e saída em kilobytes.

Os principais dispositivos de entrada e saída acessados pelo **PhotoQuality** são o disco rígido, para o carregamento dos arquivos de imagem, e a saída padrão da console, que executa o sistema e é responsável pela exibição dos resultados das análises. Os demais

dispositivos de entrada e saída, tais como teclado, *mouse* e as interfaces de rede não são acessados diretamente por esse sistema. Portanto, o volume de dados apresentado não deve levar em conta esses dispositivos.

A taxa de transferência de leitura e escrita apresentada na Figura 5.1 é possivelmente explicada por dois motivos. O primeiro poderá ser o volume de informações decorrente do carregamento dos arquivos de imagens a partir do disco rígido. O segundo motivo que pode contribuir para as taxas de transferência observadas consiste no carregamento de páginas virtuais, a partir do disco, através da técnica de paginação em disco ou *swapping*.

A média aritmética obtida entre todas as observações foi de 749,17 kB por segundo na leitura e de 0,001052 kB por segundo na escrita. O desvio padrão obtido foi de 53,91 kB para escrita e 0,00031 kB para leitura. O intervalo de confiança, utilizando uma distribuição *T de Student* com 9 graus de liberdade e com 95% de probabilidade, vai de 710,6056 até 787,7368 kBs na leitura e de 0,0008 até 0,0012 kBs na escrita.

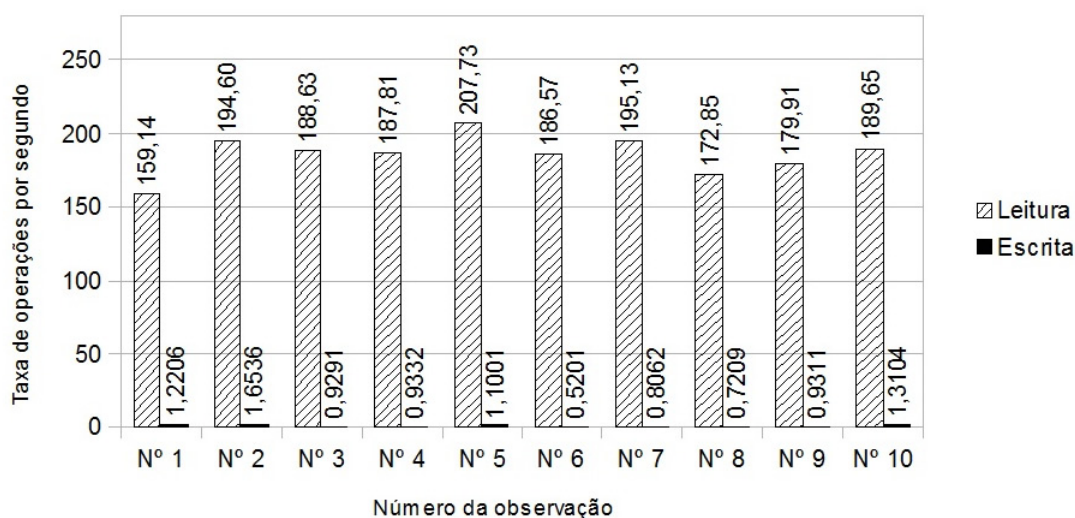


Figura 5.2: Quantidade de operações de E/S por segundo.

Como pode ser observado na Figura 5.2, as quantidades de operações de E/S por segundo realizadas pelos processos do *PhotoQuality* apresentaram um padrão semelhante aos dados referentes à taxa de transferência, apresentadas no gráfico anterior. As médias, entre as observações das taxas de operações de E/S, ficaram em 186,20 operações de leitura e 1,0125 operações de escrita por segundo, o desvio padrão ficou em 13,26 na leitura e de 0,3232 na escrita. O intervalo de confiança com 95% de probabilidade, utilizando a distribuição *T* com 9 graus de liberdade, ficou entre 176,70 e 195,70 operações de leitura por segundo e entre 0,94249 e 1,0826 operações de escrita por segundo.

Por meio dos dados apresentados, não é possível afirmar que os dispositivos de E/S estão sobrecarregados ou em estado de ociosidade. Consequentemente, não é possível



concluir que o sistema está apresentando um desempenho insatisfatório devido ao excesso de acessos aos componentes de E/S. Entretanto, não se exclui a possibilidade de haver margem para diminuição da frequência de acesso a esses dispositivos. Diminuir as quantidades de acesso aos dispositivos de E/S geralmente resulta em ganhos de desempenho importantes.

### 5.1.2 Utilização da memória RAM

Observar a utilização da memória *RAM* é relevante para verificar se uma possível falha na alocação de memória é a causadora do desempenho apresentado. O processo de consumo exacerbado de memória (*memory leak*) por um sistema poderá aumentar a frequência de paginações em disco rígido, resultando em uma quantidade maior de acessos a serem feitos, degradando, assim, sua performance e a dos demais sistemas em execução [24].

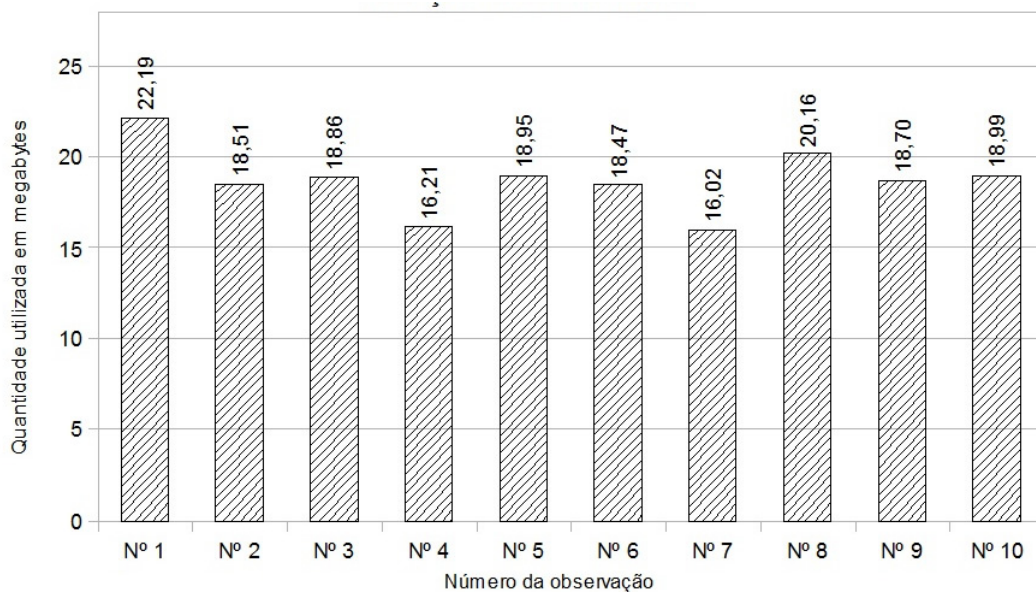


Figura 5.3: Utilização de memória RAM.

A média aritmética dos resultados de utilização do componente em questão foi de 18,7054 megabytes. O desvio padrão foi de 1,7637 e o intervalo de confiança, com base na distribuição T com 9 graus de liberdade e 95% por cento de confiança, ficou entre 17,4437 e 19,9670 megabytes.

A observação dos resultados de utilização da memória *RAM* não demonstra haver uma utilização de memória por parte do subsistema *PhotoQuality* que possa comprometer a capacidade total disponível deste recurso, cuja quantidade é de 14 GB. Além disso, com base nessa avaliação e na realizada na seção anterior (5.1.1), também não há evidências suficientes que demonstrem que o desempenho apresentado pelo sistema é resultado de um consumo excessivo de memória *RAM*, ao ponto de resultar em um aumento indireto

dos índices de paginação em disco rígido. A tendência dos dados evidencia muito mais a possibilidade de subutilização da memória disponível.

### 5.1.3 Utilização do processador

Assim como os demais componentes, o processador deve ter seu desempenho observado e avaliado. Os índices de consumo deste dispositivo podem ser uma informação chave para determinar qual a estratégia poderá ser tomada para melhorar o desempenho do sistema de modo eficaz. Caso se verifique uma utilização intensa deste dispositivo, pelo subsistema **PhotoQuality**, o conjunto de alternativas de otimização do desempenho ficarão bastante limitadas.

As alternativas disponíveis, caso a capacidade do processador se apresente como o agente limitador do desempenho, serão: alterar os algoritmos do **PhotoQuality** ou buscar acrescentar mais capacidade de processamento. Fazendo-se intervenções diretas no sistema, quebra-se uma das principais diretivas propostas no escopo do projeto, que é evitar interferir diretamente nos códigos e algoritmos deste sistema. Assim, as alternativas de ajuste serão centras no subsistema PSI.

A máquina que está hospedando a plataforma possui oito núcleos lógicos. O contador, utilizado para esta avaliação, retorna um resultado que consiste na soma das porcentagens de utilização de todos os núcleos, lógicos ou físicos, presentes na máquina. O resultado exibido, a priori, apresentou variações entre 0% e 800%. Para gerar dados que sejam mais facilmente analisáveis, os resultados apresentados são fruto de uma média aritmética feita entre o valor do contador e o número de núcleos existentes. Esse processo resultou em um número que fica no intervalo entre 0% e 100%.

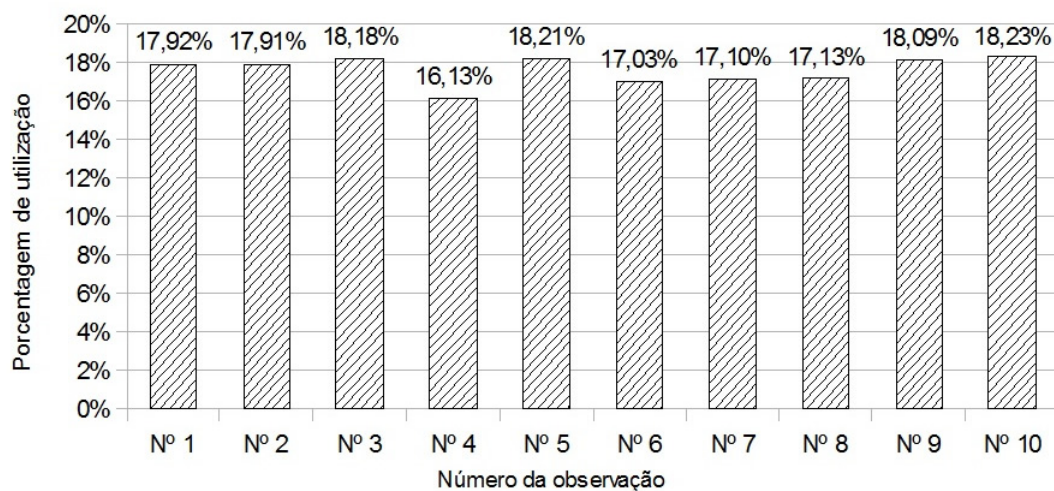


Figura 5.4: Utilização do processador.

Os dados de desempenho, ilustrados na Figura 5.4 renderam um utilização média da *CPU* de 17,60%. O desvio padrão resultante foi de 0,007066%. O intervalo de confiança, em uma distribuição T com 95% de probabilidade, ficou entre 17,09% e 18,10%.

Os dados levaram a duas possibilidades para determinação do que, de fato, está ocorrendo, para explicar a utilização observada. A primeira é que há demasiados outros processos que estão em execução no momento. A outra hipótese encontrada é que o sistema esteja executando com apenas uma instância de processamento por vez, de maneira sequencial, onde os algoritmos implementados não estão sendo capazes de extrair toda a capacidade de processamento do componente em avaliação.

A média de utilização do processador apresentada está, em geral, abaixo do desejado pelo proprietário da plataforma. Do modo como foi apresentado, o rendimento médio da *CPU*, em uma máquina dedicada exclusivamente para a avaliação das imagens, é de apenas 17,60%. O ideal é que haja uma alocação pelo sistema operacional de, no mínimo, 50% da *CPU* para realizar a tarefa à qual a máquina foi designada.

## 5.2 Hipóteses para o baixo desempenho

De posse das verificações do comportamento do hardware na execução de várias instâncias do sistema, é possível formular algumas hipóteses que expliquem o desempenho insatisfatório do sistema. Essas hipóteses serão utilizadas como ferramenta para traçar estratégias de otimização de desempenho e, assim, propor prováveis soluções ao problema que está sendo abordado.

As hipóteses a serem apresentadas são fruto da análise do comportamento individual de cada um dos dispositivos. Os gráficos de utilização de um componente individualmente podem não ser suficientes para se efetuar uma análise consistente. Por exemplo, um sistema poderá apresentar alto consumo de E/S de disco, não necessariamente ligado aos algoritmos contidos no sistema propriamente dito, mas sim devido a uma elevada utilização de memória *RAM*, que faz com que o Sistema Operacional passe a alocar excessivamente as páginas da memória virtual em disco rígido, o que resulta naturalmente em elevação da taxa de utilização deste dispositivo (*Trashing*) [25]. Portanto, a análise da taxa de utilização do disco unicamente, sem observar a memória *RAM*, ou vice-versa, não seria suficiente para chegar a uma conclusão satisfatória.

### 5.2.1 Ausência de mecanismos de concorrência

A ausência de mecanismos e/ou configurações que permitam ao subsistema PSI executar mais de uma instância de análise de imagem simultaneamente está entre as prováveis causas que estão impactando a média dos índices de vazão. Em situações onde o *hardware*

conta com mais de um núcleo de processamento, como é o caso, existe a possibilidade de aumentar a vazão observada, fazendo com que o sistema de análise de imagens tire maior proveito da capacidade do processador instalado.

Lançar mão de recursos de linhas de execução concorrentes (*threads*) também é uma alternativa a ser avaliada, já que teria o potencial de elevar a utilização do processador pelo **PhotoQuality**. Além disso, eliminaria-se o *overhead* adicional, decorrente da necessidade de uma instância individual do serviço **PSI** para atender apenas uma única requisição.

Todas as estratégias de aumento da concorrência citadas não implicam a necessidade de ter que efetuar intervenções diretas nos algoritmos do subsistema **PhotoQuality**. Ressalta-se que as diretivas estabelecidas inicialmente ainda estarão sendo obedecidas.

### 5.2.2 Leitura em disco

Apesar dos dados de leitura do disco não evidenciarem a ocorrência de ociosidade ou de sobrecarga de utilização dos dispositivos de entrada e saída, é possível que se obtenha algum ganho reduzindo ao máximo a frequência de operações de consulta ao disco rígido, já que, baseando-se na análise feita na seção 5.1.1, as operações de E/S decorrentes do carregamento de arquivos do disco rígido são provavelmente um fator preponderante para a elevação das taxas de E/S apresentada.

Existe ainda a possibilidade de que os acessos ao disco rígido possam vir a se tornar um ponto de gargalo na execução das análises quando houver diversas instâncias do processo **PhotoQuality** em execução simultaneamente, uma vez que cada instância efetuará seu conjunto de requisições de E/S individualmente e todas essas requisições somadas podem sobrecarregar o disco rígido após a implementação dos ajustes. Essa possibilidade, se confirmada, poderá gerar uma redução ainda mais acentuada dos índices de vazão, em situações em que o número de requisições estiver elevado.

### 5.2.3 Execução de um processo a cada nova requisição

A plataforma, da maneira como está implementada, executa um processo do **PhotoQuality** a cada requisição de análise de imagem. Dessa maneira, uma abordagem de execução como essa poderá exercer influência direta na performance do sistema como um todo, devido aos procedimentos que o sistema operacional precisa efetuar para possibilitar a execução de uma nova instância de um processo.

Ao iniciar uma nova instância do processo, o sistema operacional precisa consultar o disco para obter os arquivos do executável, alocar a memória e as variáveis, dentre outros procedimentos a serem efetuados. Todos esses passos impactam a performance, uma vez que demandam um tempo adicional para se preparar o ambiente de execução do processo.

Mantendo um processo ativo continuamente para realizar as análises, poderia aumentar os índices de desempenho, pois eliminaria o tempo necessário para se criar o ambiente de execução de um processo à parte, fazendo com que haja uma redução na frequência com que esse procedimento deverá ser efetuado. Assim, o tempo necessário para se efetuar a análise de uma imagem dependeria, em maior quantidade, da execução dos algoritmos implementados no sistema e menos de fatores externos que não dizem respeito diretamente à análise das imagens.

## 5.3 Soluções para o problema de desempenho

Em posse das hipóteses que expliquem as possíveis causas para o problema de desempenho apresentado pelo sistema, o próximo passo será testar se essas hipóteses são válidas. Caso um determinado ajuste resulte em aumento dos índices de desempenho, ele será implementado em definitivo no modelo de execução do sistema. A hipótese será considerada válida caso o respectivo ajuste traduza em aumento nos índices de vazão.

### 5.3.1 Estratégias de implementação dos ajustes

Para implementar os ajustes, foram definidas algumas estratégias no sentido de permitir a realização dos mesmos de modo facilitado para a equipe responsável pelo projeto de otimização do sistema. A estratégia é constituída de algumas diretivas básicas. Essas diretivas consistem em buscar fazer intervenções no subsistema **PSI**, de modo a tirar o máximo de proveito das funcionalidades da plataforma ao qual este subsistema está implementado. Em segundo lugar, está a diretiva de evitar alterações que exijam intervenção nos algoritmos e métodos do subsistema **PhotoQuality**. Intervenções no subsistema de análise das imagens serão efetuadas somente quando todas as demais alternativas de ajuste forem descartadas.

Os ajustes consistem em aumentar a utilização da *CPU*, modificando o **PSI** para que seja capaz de inicializar múltiplas instâncias do processo **PhotoQuality**, visando aumentar a vazão e escalabilidade do sistema. O segundo ajuste busca tornar o **PhotoQuality** uma biblioteca dinâmica, com o objetivo de reduzir o tempo decorrente do *overhead* que a inicialização de um processo a parte traz ao sistema como um todo. O terceiro ajuste consiste em implementar uma *RAM disk* para armazenar em memória as imagens a serem analisadas pelo sistema, visando diminuir a quantidade de operações de entrada e saída, de modo a se obter ganhos de velocidade na consulta dessas imagens.

## Implementação dos mecanismos de concorrência

O PSI é desenvolvido na plataforma *.NET* da Microsoft, por meio da linguagem **c#**. Essa plataforma conta com um *framework* abrangente para o desenvolvimento de aplicações distribuídas denominado *Windows Communication Foundation (WCF)*. O *framework* oferece métodos simplificados de ajuste nas configurações de instanciamento e concorrência do serviço. As configurações, dependendo da combinação utilizada entre elas, influenciam diretamente o número de requisições que o sistema é capaz de atender simultaneamente e, portanto, o grau de utilização dos recursos, principalmente da *CPU*.

Como o PSI é um serviço desenvolvido com base no *framework WCF*, ele poderá, a princípio, tirar proveito das configurações de instanciamento e concorrência. Instanciamento é o processo de controle do ciclo de vida dos objetos definidos para cada usuário e os demais objetos do tipo *InstanceContext*. O objeto de instância de contexto (*InstanceContext*) representa as informações de contexto para uma instância de um serviço. Concorrência refere-se ao controle do número de *threads* executando uma mesma instância de contexto, ao mesmo tempo [26].

Os comportamentos controlam como as instâncias de contexto serão criadas em resposta às mensagens recebidas e os modos de instanciamento disponíveis no *WCF* são [26]:

- **PerCall**: Uma nova instância de contexto (e, portanto, objeto do serviço) é criada a cada nova requisição do usuário.
- **PerSession**: Uma nova instância de contexto (e, portanto, objeto do serviço) é criada para cada nova sessão do cliente, sendo mantida durante toda a vida útil da sessão (isso depende de uma configuração para suporte de sessões).
- **Single**: Uma única instância de contexto é mantida e permanece responsável pela manipulação de todas as requisições dos clientes.

A concorrência controla o número de linhas de execução em uma instância de contexto, em um determinado período de tempo. Os modos de concorrência disponíveis são três [26]:

- **Single**: Cada instância de contexto permite ter, no máximo, uma *thread* no processamento de mensagens por vez.
- **Multiple**: Cada instância de contexto do serviço pode ter múltiplas *threads* efetuando o processamento de mensagens.
- **Reentrant**: Cada instância de contexto processa uma mensagem por vez, porém aceita que o próprio serviço faça uma nova requisição a si mesmo ou a um outro, criando, assim, uma nova instância de contexto.

No *WCF*, as configurações padrão de concorrência e instanciamento são *Single* e *Per-Session* [27] [28]. Em requisições TCP vindas de um mesmo nó de extremidade cliente, será tratada como uma sessão pelo serviço PSI, aproveitando-se de uma mesma conexão estabelecida anteriormente e sendo alocada uma única instância de contexto para todas as requisições subsequentes.

Isso significa que, no estado atual do sistema, é gerada apenas uma instância que contém somente uma *thread* para atender e processar todas as requisições de análise de imagens. Isso explica as observações em que foram verificadas vazões inadequadas e baixo consumo de *CPU*. Essa configuração não é ideal para favorecer a diminuição da ociosidade dos dispositivos, aumentando, assim, os índices de vazão.

Seguindo as definições, a melhor alternativa para proporcionar aumento da vazão de análises de imagens seria aquela em que se possibilita a maior vazão e com o mínimo de *overhead*. Aplicar uma configuração de instanciamento onde apenas um processo do PSI permaneceria em execução e onde o atendimento das requisições ficaria a cargo de *threads* poderia levar ao resultado desejado.

### **Conversão de processo para uma biblioteca dinâmica**

O segundo ajuste a ser efetuado será a conversão do *PhotoQuality* em um componente que possa ser instanciado e chamado por meio de métodos e não mais através da invocação de um novo processo independente para efetuar a análise. Com a chamada de métodos de uma biblioteca, espera-se eliminar o tempo gasto pelo sistema operacional na alocação de memória virtual, pilha de execução, descritores e demais recursos necessários para execução de um processo individual.

Para minimizar os desafios de implementação de um sistema, os códigos desenvolvidos até o momento serão mantidos, apenas o método de entrada e o mecanismo de compilação serão alterados para gerar uma biblioteca dinâmica (*DLL*) em que o processo ligação ocorrerá em tempo de execução. Essa estratégia procurará manter todos os métodos e algoritmos anteriormente definidos no subsistema *PhotoQuality*. O serviço do PSI fará a chamada ao subsistema através de um método de entrada, que receberá como parâmetro o endereço da imagem.

Grande parte da complexidade desse ajuste será a integração de tecnologias diferentes, onde há incompatibilidades evidentes. O PSI é desenvolvido na linguagem *c#* da plataforma *.NET*. Seu código é compilado para ser executado por uma máquina virtual denominada *Common Language Runtime (CLR)*. Em máquina virtual, é possível oferecer uma série de recursos que facilitam o desenvolvimento, tais como uma coleção extensa de bibliotecas genéricas, gerenciamento de *Threads*, *Garbage Collector*, dentre outros recursos úteis. O *PhotoQuality* foi desenvolvido na linguagem *C++* nativa, na qual existe

uma limitação nos recursos de desenvolvimento disponíveis. Assim, a linguagem conta apenas com as bibliotecas nativas ou de terceiros, onde é gerado um código executado diretamente em camadas de baixo nível, dentro do sistema operacional. Assim sendo, a comunicação entre as duas tecnologias se torna um processo relativamente complexo [3].

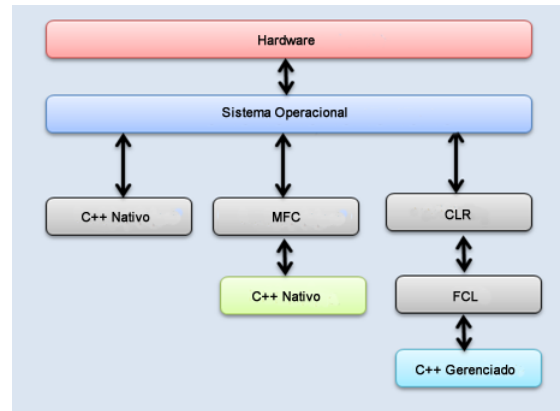


Figura 5.5: Esquema de acesso ao hardware pelo c++ nativo e pelo gerenciado [3].

Para contornar o problema da incompatibilidade entre as tecnologias, foi desenvolvida uma biblioteca intermediária (*wrapper*) em c++ gerenciado, que pode referenciar bibliotecas nativas, ao mesmo tempo em que é executada pela *CLR*. Como essa biblioteca pode ser referenciada por códigos em linguagem c#, os métodos do *PhotoQuality* podem estar disponíveis para serem chamados pelo serviço do subsistema *PSI*.

A vazão poderá apresentar melhora tendo em vista que o sistema operacional terá a possibilidade de executar o subsistema *PhotoQuality* com menor complexidade, onde será eliminada a necessidade de executar um processo a parte. Desse modo, o sistema operacional e a *CPU* ganham um tempo que poderá ser aplicado na execução de outras requisições e os índices de vazão do sistema apresentariam uma melhora considerável.

Existe ainda a possibilidade de utilizar bibliotecas estáticas, onde o processo de ligação ocorre em tempo de compilação. Esse método, em teoria, poderá traduzir em melhores índices de desempenho que o método de bibliotecas dinâmicas, onde a ligação ocorre somente em tempo de execução. Entretanto, as bibliotecas dinâmicas também são capazes de oferecer desempenho satisfatório, desde que o sistema operacional as mantenha em memória principal continuamente. Dessa forma, o desempenho permanece satisfatório sem que seja necessário abrir mão dos aspectos de modularidade, reusabilidade e manutenibilidade, que bibliotecas dinâmicas são capazes de prover.

### Minimizando o acesso ao disco rígido

O último ajuste proposto para a melhoria dos índices de vazão será a eliminação da necessidade de acesso ao disco, no carregamento das imagens a serem analisadas. Na



implementação atual, após receber uma requisição, o subsistema PSI grava o arquivo da imagem no disco. Depois, o subsistema PhotoQuality requisita ao sistema operacional a busca dessa imagem, por meio de um parâmetro que indica o endereço da imagem. Após a análise da imagem, o subsistema PSI deve remover a imagem analisada do disco. Sendo assim, no total, o procedimento demanda três acessos à memória secundária. Esse processo é repetido sistematicamente em cada nova requisição de análise.

Para evitar o custo adicional proporcionado pelos acessos ao disco rígido, cuja a velocidade de consulta é centenas de vezes inferior ao da memória principal (*RAM*) [29], é necessário utilizar uma técnica que possibilite ao subsistema PhotoQuality acessar as imagens diretamente em memória *RAM*. Isso resultaria em ganhos de desempenho relevantes, que contribuiriam diretamente para o aumento dos índices de vazão do sistema como um todo.

Seguindo a diretiva de mínima intervenção possível nos algoritmos do subsistema PhotoQuality, a alternativa mais adequada para se reduzir a quantidade de acessos à memória secundária durante as análises, seria criar um local em memória *RAM* através da técnica de *RAM disk* [30]. Utilizando essa técnica, as imagens podem ser gravadas, analisadas e deletadas nas velocidades proporcionadas pela memória principal [29]. O método de *RAM disk* permite que os dados sejam gravados através de um sistema de arquivos próprio desenvolvido para armazenamento em memória *RAM* [31]. Utilizando um sistema de arquivos, as imagens podem ser consultadas como se estivessem normalmente armazenadas no disco rígido. E assim, podem ser acessadas sem qualquer modificação nos algoritmos do subsistema de análise de imagens.

A técnica de (*RAM disk*) é desenvolvida para aparentar como qualquer outro periférico de disco conectado à máquina. A interface entre o processador e a unidade, baseada em memória, é mantida por meio de um barramento de E/S, e o acesso é semelhante a outros dispositivos de armazenamento magnéticos. Em vez de armazenar os dados em discos magnéticos, o *RAM disk* armazena as informações em um grande vetor de dados na *RAM*. Assim, a latência de acessar o *RAM disk* é praticamente zero, se comparada com os 15 a 50 milissegundos de latência dos discos magnéticos. *RAM disks* também possuem a funcionalidade de poderem ser consultadas e manipuladas em alta velocidade por outros dispositivos, sem a necessidade de intermediação da *CPU* através do método de acesso direto à memória (*DMA*) [31].

Este capítulo levantou e descreveu algumas hipóteses para o baixo desempenho que motivou este projeto. Além disso, apresentou algumas estratégias para otimizar o sistema, com base nas hipóteses levantadas. No capítulo seguinte, os ajustes serão implementados e os resultados apresentados e avaliados, para atestar se estes otimizaram o sistema significativamente.

## Capítulo 6

# Apresentação e avaliação dos resultados

Neste capítulo, o enfoque será no teste das hipóteses apresentado no capítulo anterior. Para testar a eficácia dos ajustes, primeiramente serão coletadas as métricas de desempenho apresentadas pelo sistema no momento em que se encontram atualmente. Posteriormente, será feita a implementação dos ajustes sugeridos no modelo implementado. Ao fim da implementação, serão coletadas novamente as métricas para serem comparadas (teste de *benchmarking*) com as métricas anteriores.

Foram estabelecidos cenários de desempenho para o melhor acompanhamento da otimização e avaliação do desempenho. O primeiro cenário visa observar o desempenho do sistema sem qualquer otimização. Já os próximos três cenários objetivam a implementação e avaliação dos ajustes propostos resultaram em melhoras significativas no desempenho do sistema.

- **Cenário I** – Sistema de análise de imagens na primeira implementação, sem *tuning* ou qualquer outra estratégia de otimização do despenho.
- **Cenário II** – Implementação do PSI com apenas uma instância de execução do subsistema e com múltipla concorrência de processamento das requisições. *InstanceMode: Single* e *ConcurrencyMode: Multiple*.
- **Cenário III** – Execução do subsistema *PhotoQuality* como uma biblioteca dinâmica.
- **Cenário IV** – Gravação e carregamento das imagens a serem analisadas por meio de um sistema de arquivos em *RAM disk*.

Cada cenário é composto de um ajuste e um teste de desempenho subsequente, exceto no Cenário I, no qual não há ajuste implementado. Portanto, esse cenário conta apenas

com o teste de desempenho para detectar, com a máxima precisão, qual é o desempenho apresentado pelo sistema inicialmente. Em cada cenário, os testes foram organizados da seguinte maneira: definição de oito amostras com cargas de trabalho de imagens diferentes; cada amostra é submetida a dez observações distintas, onde a carga de trabalho é composta por imagens tomadas aleatoriamente, sendo obtidas medidas das taxas de vazão para cada uma dessas observações; posteriormente, é calculada a média de vazão obtida em cada observação, além do cálculo da média de todas as observações.

As médias dos resultados do cenário em questão serão comparadas com o cenário anterior, desde que o anterior tenha resultado em melhoras significativas no desempenho. A comparação entre os cenários será feita entre as amostras de cada cenário cujo tamanho é o maior disponível, ou seja, as amostras de 12800 imagens.

A validação de um cenário ocorrerá por meio de um teste T com duas amostras de variâncias presumidamente diferentes, já que o conjunto de imagens utilizadas sofre alteração de um teste para outro [32]. O nível de significância dos testes foi fixado em 0,05.

O monitor utilizado nos testes foi desenvolvido exclusivamente para este projeto. É voltado para monitorar apenas os índices de vazão do sistema e salvar os dados obtidos em arquivo de texto. O algoritmo basicamente dispara um grupo de 100 requisições simultâneas, com imagens aleatórias, para serem avaliadas até que se complete a quantidade de requisições estipuladas para o teste. Ao fim, o tempo total gasto é contabilizado em milissegundos para a conclusão do teste.

Esse monitor assimila as características de um monitor *online*, em que sua ativação ocorre quando o monitoramento anterior se encerrou. Como esse monitor executa em uma máquina a parte, a princípio, o *overhead* gerado no servidor de análise de imagens pode ser desconsiderado.

A carga utilizada nos testes provém de um conjunto de 39.000 imagens enviadas por indivíduos, para fins de identificação pessoal. Essas imagens foram cedidas pela instituição que patrocina o projeto, para a realização dos testes de desempenho. Para os testes, as imagens serão selecionadas aleatoriamente.

## 6.1 Cenário I – Medições de desempenho iniciais

O gráfico da Figura 6.1 apresenta que o sistema, sem qualquer otimização, é capaz de processar e responder cerca de duas requisições de análises de imagens por segundo. O tamanho da carga de imagens utilizada apresenta influências na taxa de análises até certo ponto. Até uma carga de 800 imagens, ainda é possível perceber uma variação considerável nas taxas médias de análise por segundo. Isso evidencia que as características de cada

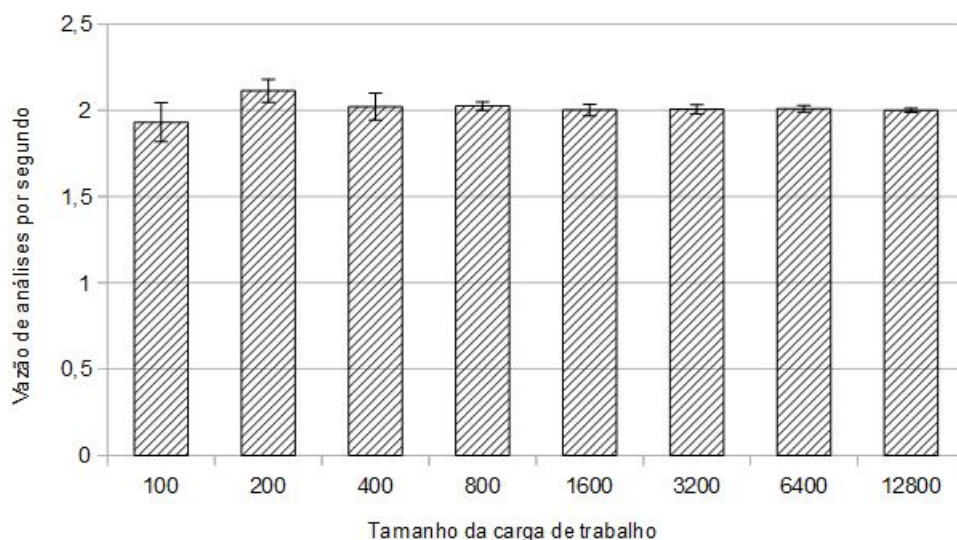


Figura 6.1: Médias de vazão por amostra e seus erros estimados, dentro do Cenário I.

imagem influenciam no tempo de resposta e, conseqüentemente, na vazão do sistema. Porém, a partir de um certo tamanho de carga, entre 800 e 1600 imagens, as taxas de vazão tendem a se estabilizar cada vez mais em duas análises por segundo.

As barras de erro estimadas em cada conjunto de carga apresentam os limites à esquerda e à direita para o valor da média de taxas de vazão, com 95% de confiança em uma distribuição T de Student. Quanto menor o comprimento da barra, menor é o erro esperado para a média e, conseqüentemente, maior a precisão do valor da média de vazão para a amostra.

Tabela 6.1: Resultados dos testes desempenho no Cenário I

	Tamanho	Média	Desvio padrão	erro
Amostra 1	100	1,9305868966	0,1575286753	0,1126813967
Amostra 2	200	2,1128985146	0,0935367002	0,0669074758
Amostra 3	400	2,0217291182	0,1098984507	0,0786111538
Amostra 4	800	2,0246136007	0,0339747541	0,0243023865
Amostra 5	1600	2,0015018584	0,0466669488	0,0333812048
Amostra 6	3200	2,005811834	0,036074053	0,0258040301
Amostra 7	6400	2,0101008164	0,0260147054	0,0186085063
Amostra 8	12800	2,0005658081	0,01567574	0,0112129698

Na Tabela 6.1, é possível visualizar os dados utilizados para gerador nos testes de vazão do sistema. São apresentadas as médias em cada uma das amostras, bem como o desvio padrão e a estimativa de erro com 95% de confiança para a média.

Tomando como referência a amostra de 12800 imagens, cuja média registrada foi de duas imagens por segundo, é possível concluir que o sistema, na configuração de funcionamento atual onde não há qualquer otimização implementada, não é capaz de oferecer um desempenho minimamente próximo ao objetivo definido inicialmente.

## 6.2 Cenário II – Ajuste dos parâmetros de instanciamento e concorrência

Neste cenário, os ajustes efetuados restringiram a apenas uma instância do serviço PSI, com a possibilidade iniciar múltiplas linhas de execução, uma para cada requisição. Esses parâmetros foram selecionados com base nas definições de instanciamento e concorrência presentes na plataforma *WCF*. Com essa implementação, haverá apenas uma instância de execução do subsistema PSI, mas com suporte a múltiplas *threads* para casos em que há uma elevada demanda pelo serviço. Pela definição, essa configuração proporcionará o melhor desempenho, visto que a quantidade de instâncias de processos do PSI será limitada a apenas uma, porém com a possibilidade prover escalabilidade mediante a criação de mais linhas de execução concorrentes quando necessário.

O fabricante do *framework WCF* não especifica publicamente os detalhes técnicos referentes à estratégia de implementação da concorrência por meio de *threads*. Desse modo, não é possível explicitar se as *threads* lançadas pelo *WCF* utilizam técnicas convencionais de implementação dessas entidades. Consequentemente, também não é possível verificar se o *framework* possui uma abordagem de linhas de execução leves ou *Tasks* por meio da biblioteca *Task Parallel Library (TPL)*.

Tabela 6.2: Resultados das vazões obtidas no Cenário I e no Cenário II

Tamanho	Cenário II			Cenário I		
	Média	Desvio padrão	Erro	Média	Desvio padrão	Erro
100	5,5810	0,5843	0,4179	1,9305	0,1575	0,1126
200	6,0677	0,1972	0,1411	2,1128	0,0935	0,0669
400	6,6790	0,2740	0,1960	2,0217	0,1098	0,0786
800	6,6852	0,3454	0,2470	2,0246	0,0339	0,0243
1600	6,6193	0,2045	0,1463	2,0015	0,0466	0,0333
3200	6,6989	0,1022	0,0731	2,0058	0,0360	0,0258
6400	6,6341	0,1900	0,1359	2,0101	0,0260	0,0186
12800	6,5473	0,0633	0,0453	2,0005	0,0156	0,0112

Na Tabela 6.2, é possível visualizar os dados utilizados para gerador nos testes de vazão do sistema. São apresentadas as médias em cada uma das amostras, bem como o

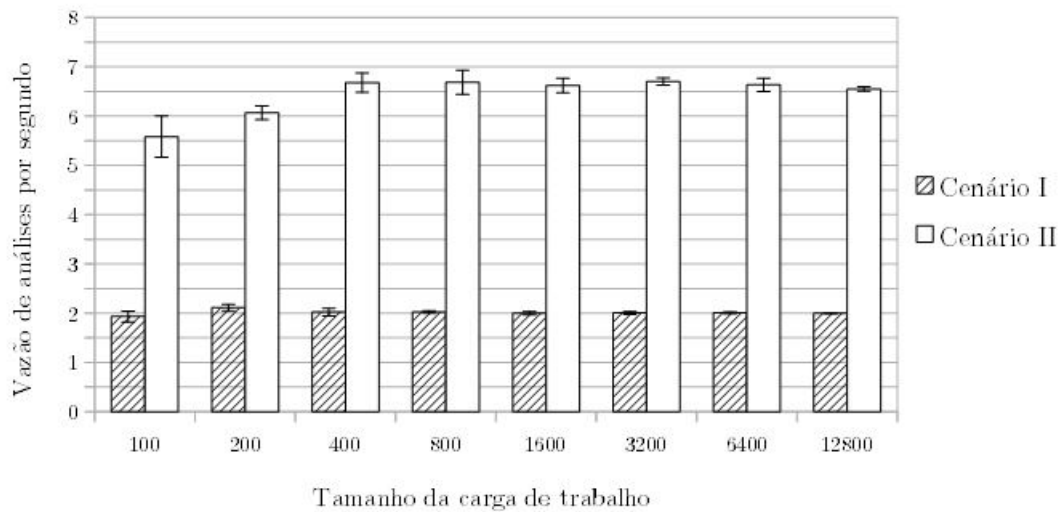


Figura 6.2: Médias de vazão por amostra e seus erros estimados, no Cenário I e II.

desvio padrão e a estimativa de erro com 95% de confiança para a média. As médias de vazão partem de 5,5 análises por segundo no cenário II, estabilizando-se em de 6,6 análises por segundo nas amostras de tamanhos maiores que 200 imagens. Nesse cenário, o desvio padrão não parece estar ligado diretamente ao tamanho da carga utilizada. De modo que apresentou aumento ou diminuição, em relação as observações anteriores de menor carga. Mas em geral é possível ser observado uma redução significativa do intervalo de erro esperado para a média entre o primeiro e o último conjunto de cargas. O intervalo de variação esperado para a média acompanha o comportamento do desvio padrão de cada amostra.

O gráfico da Figura 6.2 apresenta os valores das médias de vazões obtidas entre o cenário I e o II. As barras de erro apresentam a estimativa dos limites inferiores e superiores para a média, com 95% de confiança. De modo geral, como pode ser observado no gráfico, o ajuste representou melhoras evidentes de aumento da vazão do sistema. A comparação, entre as médias das amostras com 12.800 imagens, permitiu chegar à conclusão de que houve cerca de 320% de aumento nos índices de vazão, entre os dois cenários.

### 6.2.1 Conclusão ajustes

#### Teste de hipóteses: uni-caudal à direita.

$H_0$  : Os índices de vazão obtidos no cenário I e no cenário II não diferem significativamente.

$H_1$  : Os índices de vazão tiveram um aumento significativo após os ajustes implementados no cenário II.

Tabela 6.3: Resultado do teste T com duas amostras e variâncias presumidamente diferentes, utilizando as observações com 12.800 imagens.

	Cenário II	Cenário I
Média	6,54733852	2,0005658081
Desvio padrão	0,0633911716	0,01567574
Observações	10	10
Hipótese da diferença de média	0	
Graus de liberdade	10	
Valor do $t$	220,1841317189	
$P(T \leq t)$ uni-caudal (p-valor)	0,00000	
$t$ crítico uni-caudal	1,8124611228	

Por meio dos valores presentes na Tabela 6.3, deve-se rejeitar a hipótese  $H_0$  e aceitar a  $H_1$ , pois o valor do  $t$  calculado é maior que o valor de  $t$  crítico uni-caudal. Além disso, o resultado do p-valor é inferior ao nível de significância estipulado de 0,05. Ou seja, o ajuste no instanciamento e concorrência resultou em aumento significativo nos índices de vazão do sistema.

### 6.3 Cenário III – Execução do PhotoQuality como biblioteca dinâmica

Este cenário consiste em implementar e observar o desempenho do subsistema PhotoQuality como uma biblioteca dinâmica. Nesse cenário, os ajustes efetuados anteriormente permanecerão no sistema. Desse modo, as comparações de desempenho serão feitas em relação aos índices de vazão obtidos no cenário II.

A implementação mediante biblioteca dinâmica foi escolhida devido às facilidades de reúso que esse método proporciona. Desse modo, é possível efetuar ajustes no subsistema PSI sem a necessidade recompilar também o subsistema PhotoQuality e vice-versa. A abordagem mediante bibliotecas estáticas é mais eficiente em casos onde os trechos de códigos definidos nessas bibliotecas não são demandados de maneira contínua. Porém, espera-se que o subsistema de análise de imagens deverá permanecer em execução continuamente quando o serviço estiver em produção, fazendo com que a implementação como biblioteca dinâmica seja uma alternativa mais adequada.

Novamente, para fins de comparação, as amostras selecionadas foram as que possuem maior número de imagens avaliadas (amostras com 12.800 imagens). Em primeira observação, o gráfico da Figura 6.3 apresenta valores das médias de vazão onde nem todos os tamanhos de carga (em especial, os tamanhos com 3.200 e 6.400 imagens) apresentaram as

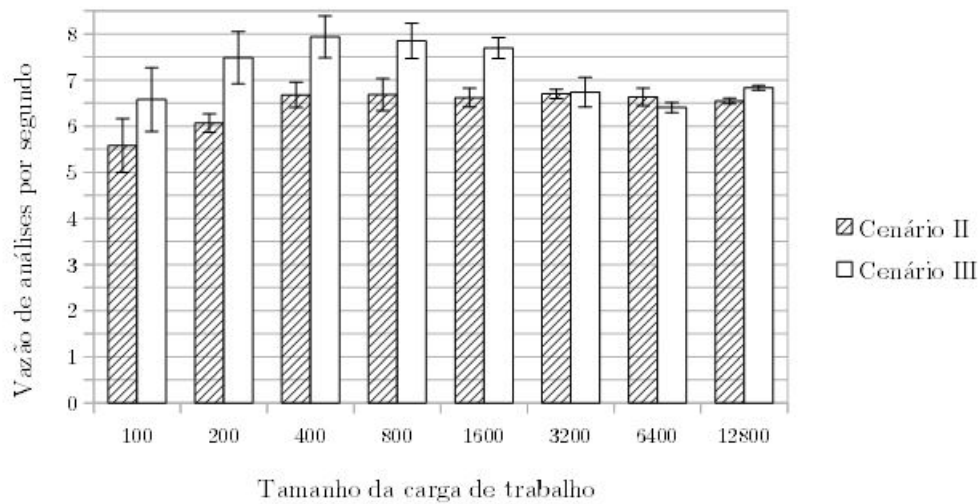


Figura 6.3: Médias de vazão por amostra e seus erros estimados, no cenário II e III.

médias de desempenho que corroboram a hipótese de melhora do desempenho no cenário III. Porém, levando-se em consideração os valores das estimativas de erro calculadas para as médias, ainda é possível perceber que houve uma melhora significativa de desempenho também nas amostras com os tamanhos em questão. As demais amostras confirmaram que os ajustes realizados nesse cenário resultaram em aumento no desempenho do sistema de análise de imagens.

Os resultados apresentados na Figura 6.3 – em que as médias de vazão com amostras de 3.200 e 6.400 imagens, no cenário III, aparentam ser inferiores ao cenário anterior – podem ser decorrentes, em geral, de fatores tais como: procedimentos do sistema operacional que demandaram *CPU* no instante em que as amostras de carga em questão estavam sendo analisadas; o *PhotoQuality* apresentou falhas de alocação de memória, no momento de execução destas análises, que fizeram com que houvesse um consumo exacerbado de memória *RAM*, o que pode ter levado ao *trashing*, resultando no aumento das operações de disco e, conseqüentemente, queda da vazão do sistema. Todas essas hipóteses deverão passar por avaliação futuramente, fora do escopo deste trabalho, quando deverão ser observados todos fatores envolvidos no momento da análise.

### 6.3.1 Conclusão dos ajustes

#### Teste de hipóteses: uni-caudal à direita.

$H_0$  : Os índices de vazão obtidos no cenário II e no cenário III não diferem significativamente.



Tabela 6.4: Resultados das vazões obtidas no cenário III e no cenário II

Tamanho	Cenário III			Cenário II		
	Média	Desvio padrão	Erro	Média	Desvio padrão	Erro
100	6,5788	0,6910	0,4179	5,5810	0,5843	0,4180
200	7,4855	0,5670	0,1411	6,0677	0,1973	0,1411
400	7,9378	0,4532	0,1960	6,6790	0,2740	0,1960
800	7,8520	0,3803	0,2470	6,6853	0,3454	0,2471
1600	7,6953	0,2226	0,1463	6,6194	0,2046	0,1463
3200	6,7353	0,3211	0,0731	6,6989	0,1023	0,0732
6400	6,4035	0,1111	0,1359	6,6341	0,1901	0,1359
12800	6,8335	0,0507	0,0453	6,5473	0,0634	0,0453

Tabela 6.5: Resultado do teste T com duas amostras e variâncias iguais, utilizando as observações com 12.800 imagens.

	Cenário III	Cenário II
Média	6,8452456101	6,54733852
Desvio padrão	0,0667965299	0,0633911716
Observações	10	10
Hipótese da diferença de média	0	
Graus de liberdade	18	
Valor do $t$	10,2300405597	
$P(T \leq t)$ uni-caudal (p-valor)	0,00000	
$t$ crítico uni-caudal	1,7340636066	

$H_1$  : Os índices de vazão tiveram um aumento significativo após os ajustes implementados no cenário III.

Por meio dos valores presentes na Tabela 6.5, deve-se rejeitar a hipótese  $H_0$  e aceitar a  $H_1$ , pois o valor do  $t$  calculado é maior que o valor de  $t$  crítico uni-caudal. Além disso, o resultado do p-valor é inferior ao nível de significância estipulado de 0,05. Ou seja, o sistema obteve ganhos de desempenho significativos com o PhotoQuality funcionando como uma biblioteca dinâmica.

## 6.4 Cenário IV — Gravação e leitura das imagens em *RAM disk*

O cenário IV consiste na modificação dos mecanismos de salvamento e carregamento das imagens a serem analisadas. Nos ajustes, foi definido um sistema de arquivos em um *RAM disk*. O sistema de arquivos escolhido na implementação foi *New Technology File*

*System (NTFS)* por ser o sistema já implementado pelo sistema operacional utilizado e, em geral, consumir menos memória RAM, em comparação ao sistema de arquivos alternativos disponíveis para sistemas operacionais *Windows*.

Para montar o *RAM disk*, utilizou-se o *software SoftPerfect RAM Disk®*, pois foi a alternativa gratuita que proporciona os melhores desempenho dentre as demais alternativas disponíveis [33]. Os testes de desempenho – que foram realizados para determinar o *software* com melhor velocidade de leitura e escrita – foram efetuados utilizando o *software* de *benchmark* de discos *CrystalDiskMark®*.

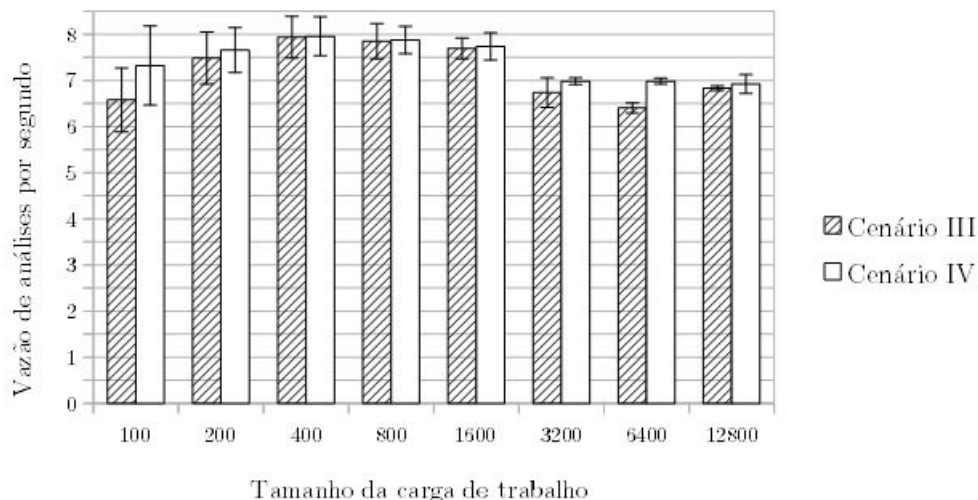


Figura 6.4: Médias de vazão por amostra e seus erros estimados no cenário III e IV.

A Figura 6.4 exibe as médias de vazão obtidas no cenário atual e no anterior. Como pode ser percebido, na maioria das cargas utilizadas, o sistema não obteve ganhos na vazão após o ajuste proposto. É possível observar que, nas observações com cargas entre 200 e 1.600 imagens, as médias de vazão são semelhantes entre os dois cenários avaliadas. A primeira vista, as observações com 100 imagens apresentaram aumento significativo também, porém as barras de erro não permitem chegar à conclusão de que um aumento significativo ocorreu, de fato, no sistema, com uma carga desse tamanho.

Também é possível observar, ainda na Figura 6.4, que apenas as observações, cujas cargas possuem 6.400 imagens, obtiveram ganhos de vazão expressivos. Porém, conforme discutido nas seções anteriores, existem fortes indícios de o sistema ter sofrido intervenções de rotinas de outros processos que inicializaram no momento das análises ou que possa ter havido problemas na alocação de memória que influenciaram as médias de vazão apresentada. A Tabela 6.6 apresenta os dados utilizados para as análises.

É interessante notar que as observações com 6.400 imagens apresentaram um resultado oposto aos obtidos nas observações com o mesmo tamanho realizadas no cenário

anterior (Seção 6.3), onde as médias de vazão para as observações com 6.400 imagens foram inferiores às das demais amostras. No caso do cenário IV, os dados apresentaram resultados em que foi observado um aumento expressivo da vazão em relação ao cenário anterior. Isso faz com que se fortaleça a hipótese de que houve um consumo exacerbado de memória *RAM* por algum processo (possivelmente o próprio **PhotoQuality**), durante as observações com cargas de tamanho 6.400, já que, quando há aumento da utilização de memória *RAM* muito acima do normal, o sistema operacional passa a fazer alocação das páginas virtuais em disco, gerando sobrecarga nesse dispositivo. Após o ajuste proposto no cenário atual (onde há maior independência do disco, uma vez que as imagens são carregadas diretamente da memória *RAM*), o sistema deve ter sentido menos o impacto da sobrecarga de utilização do disco, resultando em um aumento no desempenho.

Tabela 6.6: Resultados das vazões obtidas no cenário IV e no cenário III

Tamanho	Cenário IV			Cenário III		
	Média	Desvio padrão	Erro	Média	Desvio padrão	Erro
100	7,3247	1,2012	0,8592	6,5787	0,9660	0,6910
200	7,6561	0,6791	0,4857	7,4854	0,7926	0,5669
400	7,9579	0,5860	0,4192	7,9378	0,6335	0,4531
800	7,8759	0,4138	0,2960	7,8519	0,5316	0,3802
1600	7,7368	0,4112	0,2941	7,6952	0,3112	0,2226
3200	6,9827	0,1028	0,0735	6,7352	0,4489	0,3211
6400	6,9855	0,0855	0,0612	6,4035	0,1553	0,1111
12800	6,9249	0,2849	0,2038	6,8335	0,0708	0,0506

### 6.4.1 Conclusão dos ajustes

Tabela 6.7: Resultado do teste T com duas amostras e variâncias presumidamente diferentes, utilizando as observações com 12.800 imagens.

	Cenário IV	cenário III
Média	6,9249934231	6,8452456101
Desvio padrão	0,2849302691	0,0667965299
Observações	10	10
Hipótese da diferença de média	0	
Graus de liberdade	10	
Valor do $t$	0,861712943	
$P(T \leq t)$ uni-caudal (p-valor)	0,2045166184	
$t$ crítico uni-caudal	1,8124611228	

**Teste de hipóteses: uni-caudal à direita.**

$H_0$  : Os índices de vazão obtidos no cenário IV e no cenário III não diferem significativamente.

$H_1$  : Os índices de vazão tiveram um aumento significativo após os ajustes implementados no cenário IV.

Por meio dos valores presentes na Tabela 6.7, deve-se aceitar a hipótese  $H_0$  e rejeitar a  $H_1$ , pois o valor do  $t$  calculado é menor que o valor de  $t$  crítico uni-caudal. Além disso, o resultado do p-valor é superior ao nível de significância estipulado de 0,05. Ou seja, o sistema não obteve melhora nas médias de vazão após a implementação da técnica de armazenamento das imagens em um sistema de arquivos baseado em *RAM disk*.

Este capítulo apresentou os resultados dos ajustes de propostos, além de ter feito as discussões e citado as implicações pertinentes em cada um dos ajustes. O próximo capítulo apresentará as conclusões do trabalho, além de sugerir novas etapas para o aperfeiçoamento do projeto.

# Capítulo 7

## Conclusões

Este trabalho demonstrou que é possível integrar dois sistemas de plataformas distintas, de maneira a não penalizar o desempenho da plataforma. Conforme apresentado nos ajustes, pode-se obter um aumento considerável nas médias de vazão apenas ajustando-se os parâmetros de instanciamento e concorrência do serviço. Também demonstrou-se o quanto o desempenho pode ser melhorado com a conversão de um processo a parte em um componente modular integrado à solução mediante biblioteca dinâmica.

Além de prover um melhor desempenho ao sistema de análise de imagens, este projeto também demonstrou que – em casos onde apenas deseja-se aumentar a capacidade de vazão e escalabilidade de sistemas computacionais – é possível otimizar um sistema mais facilmente sem necessariamente haver uma intervenção direta nos algoritmos internos, bastando apenas a modificação dos parâmetros ou do modelo de execução para que um objetivo seja, de certa forma, atingido.

A maioria dos ajustes propostos neste trabalho resultou em efeitos positivos no desempenho. Cabe ressaltar que, caso a análise de desempenho tivesse permeado o desenvolvimento do sistema **PhotoQuality** desde as etapas iniciais do projeto, os resultados obtidos nos ajustes poderiam ter sido ainda mais significativos. Além de prover melhores resultados, pensar na questão do desempenho o quanto antes, possibilita detectar e corrigir, de imediato, defeitos no *software* que possam vir a ser introduzidos após ajustes de desempenho. Por causa disso, a questão de desempenho deve ser levada em consideração desde as etapas de iniciais de desenvolvimento de sistemas computacionais.

A Figura 7.1 apresenta o desempenho obtido em cada cenário. É possível perceber com clareza a melhora no desempenho no decorrer da implementação dos ajustes. Os ganhos mais significativos ocorreram entre o cenário I e III, onde as médias de vazão foram significativamente aumentadas após a implementação dos respectivos ajustes.

O gráfico na Figura 7.1 mostra que a melhor vazão média obtida foi de cerca de 8,5 imagens por segundo, resultado que fica acima da média de vazão estipulada pelo

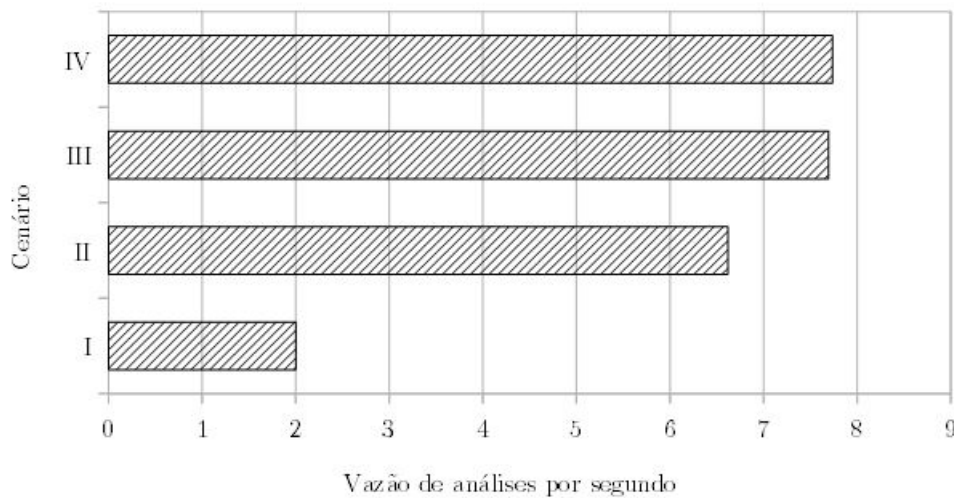


Figura 7.1: Médias de vazão por Cenário, utilizando-se as observações com 1.600 imagens.

patrocinador do projeto, que foi de 5,4 imagens por segundo. Porém, em nenhum momento durante o decorrer dos ajustes e testes de desempenho, foi possível obter um pico de vazão onde foram processadas 14 imagens por segundo, como desejado pelo patrocinador.

Os valores apresentados não invalidam os resultados obtidos, uma vez que ainda podem ser obtidas até 17 análises em um intervalo de dois segundos. Desse modo, a requisição mais lenta do sistema poderá levar até dois segundos para ser processada e respondida. Caso o proprietário do projeto considere que esse intervalo fere os Acordos de Níveis de Serviço (*SLAs*), outra estratégia poderá ser adotada utilizando o **planejamento da capacidade**, onde podem ser avaliadas as alternativas de aumento da capacidade de processamento do servidor ou a alocação de uma máquina física exclusivamente para a hospedagem do serviço.

## 7.0.2 Trabalhos futuros

No decorrer dos ajustes feitos no cenário III, onde **PhotoQuality** foi implementado como biblioteca dinâmica, detectou-se a diminuição dos índices de vazão nas observações com mais de 1.600 imagens. A hipótese de que surgiu um defeito no software durante a implementação da biblioteca dinâmica é bastante plausível. Esse problema pode ter aumentando consideravelmente o consumo de memória principal, o que fez com que o sistema operacional passasse a alocar uma quantidade maior de paginas de virtuais em disco rígido.

Dessa forma, os trabalhos futuros devem ser voltados para a resolução do problema em questão, de modo a não impactar a escalabilidade do sistema quando for colocado em produção posteriormente.

É possível que haja maiores ganhos de desempenho ao se testar novas combinações de ajustes no instanciamento e na concorrência, de modo que se encontre uma combinação entre esses parâmetros que seja ainda mais adequada para prover índices de vazão do sistema ainda mais favoráveis.

Ao finalizar com sucesso os testes e ajustes de desempenho, a ferramenta agora deverá ser implementada na infraestrutura destinada à hospedagem. No modelo de execução escolhido, a fase de implementação deverá ser um processo bastante simples, onde se estima que seja necessária apenas a substituição do **PhotoQuality** pela nova versão desenvolvida, na qual o sistema será referenciado pelo PSI como uma biblioteca adicional, junto ao servidor destinado a hospedar a plataforma. As etapas da implementação consistirão em:

- Compilar a plataforma e todas as suas bibliotecas em modo *release*.
- Instalação do produto no servidor de produção.
- Configurar o PSI para referenciar o componente **PhotoQuality**.
- Treinar a equipe de infraestrutura para manutenções preventivas, caso seja necessário.
- Disponibilização do serviço no ambiente de produção.

# Referências

- [1] W.W. Osterhage. *Computer Performance Optimization: Systems - Applications - Processes*. SpringerLink : Bücher. Springer Berlin Heidelberg, 2014. xi, 13
- [2] Document 9303 machine readable travel documents (mrtcd). volume 1, part 1: Machine readable passports. xi, 17, 19, 22, 23, 24, 25, 26
- [3] Microsoft. Managed c++/cli programming. Disponl em <http://resources.infosecinstitute.com/managed-ccli-programming-part-1/>. xi, 51
- [4] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley professional computing. Wiley, 1991. xiii, 6, 7, 8, 9, 11, 12
- [5] Nenad Medvidovic, David S. Rosenblum, and Richard N. Taylor. A language and environment for architecture-based software development and evolution. In *Proceedings of the 21st International Conference on Software Engineering, ICSE '99*, pages 44–53, New York, NY, USA, 1999. ACM. 1
- [6] W3C. Web services description language (wsdl) 1.1. <http://www.w3.org/TR/wsdl>. 2
- [7] Ian Molyneaux. *The Art of Application Performance Testing: Help for Programmers and Quality Assurance*. O'Reilly Media, Inc., 1st edition, 2009. 2
- [8] David A. Patterson and John L. Hennessy. *Computer Organization and Design, Revised Fourth Edition, Fourth Edition: The Hardware/Software Interface*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edition, 2011. 3
- [9] Institute of Electrical and CORPORATE Electronics Engineers, Inc. Staff. *IEEE Software Engineering Standards Collection, Spring 1991*. IEEE Standards Office, New York, NY, USA, 1991. 5
- [10] Connie U. Smith. *Performance Engineering of Software Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1990. 5
- [11] M.S. Obaidat and N.A. Boudriga. *Fundamentals of Performance Evaluation of Computer and Telecommunications Systems*. Wiley, 2010. 6, 8, 11
- [12] J.Y.L. Boudec. *Performance Evaluation of Computer and Communication Systems*. Computer and communication sciences. EFPL Press, 2010. 6



- [13] K. Kant. *Introduction to computer system performance evaluation*. McGraw-Hill computer science series. McGraw-Hill, 1992. 6, 7, 9, 11, 12
- [14] D. Ferrari. Workload characterization and selection in computer performance measurement. *Computer*, 5(4):18–24, July 1972. 10
- [15] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. Wiley Publishing, 8th edition, 2008. 13, 14, 15
- [16] Milan Stevanovic. *Advanced C and C++ Compiling*. Apress, Berkely, CA, USA, 1st edition, 2014. 15
- [17] George I. Davida and Yvo G. Desmedt. Refereed articles: Passports and visas versus ids. *Comput. Secur.*, 11(3):253–258, may 1992. 20
- [18] G. I. Davida and Y. G. Desmedt. Passports and visas versus ids. In *Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT’88*, pages 183–188, New York, NY, USA, 1988. Springer-Verlag New York, Inc. 20
- [19] Document 9303 machine readable travel documents (mrttd). volume 3, part 1: Machine readable passports. <http://www.icao.int/publications/pages/publication.aspx?docnum=9303>. 20
- [20] U.S. Passport Service Guide. E-passport: All about the electronic passport. <http://www.us-passport-service-guide.com/e-passport.html>. 20
- [21] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. 29
- [22] Microsoft. What is windows communication foundation. Disponl em <https://msdn.microsoft.com/en-us/library/ms731082.aspx>. 34
- [23] Microsoft TechNet. Monitor de desempenho do windows. <https://technet.microsoft.com/pt-br/library/cc749249.aspx>. 40
- [24] Memory leak detection in embedded systems. 44
- [25] Peter J. Denning. Thrashing: Its causes and prevention. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS ’68 (Fall, part I), pages 915–922, New York, NY, USA, 1968. ACM. 46
- [26] Microsoft. Sessions, instancing, and concurrency. Disponl em <https://msdn.microsoft.com/en-us/library/ms731193>. 49
- [27] Microsoft Development Network (MSDN). Servicebehaviorattribute.concurrency mode property. <https://msdn.microsoft.com/en-us/library/system.servicemodel.servicebehaviorattribute.concurrency mode>. 50
- [28] Microsoft Development Network (MSDN). Servicebehaviorattribute.instancecontextmode property. <https://msdn.microsoft.com/en-us/library/system.servicemodel.servicebehaviorattribute.instancecontextmode>. 50

- [29] Tobias Kind. Ramdisk benchmarks. *University of California*, 2011. 52
- [30] D. McKinley. Power efficient ram disk and a method of emulating a rotating memory disk, February 20 1996. US Patent 5,493,574. 52
- [31] Marshall K McKusick, Michael J Karels, and Keith Bostic. A pageable memory based filesystem. *USENIX Summer*, 1990. 52
- [32] C. Werkema. *Inferência Estatística: Como Estabelecer Conclusões Com Confiança No Giro Do Pdca E Dmaic*. Elsevier Brasil. 54
- [33] 12 ram disk software benchmarked for fastest read and write speed. <https://www.raymond.cc/blog/12-ram-disk-software-benchmarked-for-fastest-read-and-write-speed/>. 61